



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 141599**

**RANCANG BANGUN SISTEM MONITORING  
PEMAKAIAN AIR MENGGUNAKAN KOMUNIKASI  
MULTIHOP PADA JARINGAN SENSOR NIRKABEL**

Fadli Rahmawan  
NRP 2211 100 074

Dosen Pembimbing  
Dr. Ir. Wirawan, DEA.  
Dr. Istas Pratomo, ST., MT.

JURUSAN TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015



**FINAL PROJECT - TE 141599**

***Visual Sensor Network with Energy Saving Algorithm***

Hananto Agung Baskoro  
NRP 2211 100 150

Lecture Advisor  
Eko Setijadi ST., MT., Ph.D  
Dr. Ir. Wirawan, DEA

ELECTRICAL ENGINEERING MAJOR  
Industry Technology Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2015

**VISUAL SENSOR NETORKDENGANALGORITMAYANG  
HEMAT ENERGI**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik Elektro  
pada  
Bidang Studi Telekomunikasi Multimedia  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember**

**Menyetujui**

**Dosen Pembimbing I,**

**Dosen Pembimbing II,**



**Eko Setijadi, ST., MT., Ph.D**  
**NIP. 196510141990021001**



**Dr. Ir. Wirawan, DEA.**  
**NIP. 195506221987011001**



# **VISUAL SENSOR NETWORK DENGAN ALGORITMA YANG HEMAT ENERGI**

Hananto Agung Baskoro  
2211 100 150

Dosen Pembimbing 1 : Eko Setijadi, ST., MT., Ph.D  
Dosen Pembimbing 2 : Dr. Ir. Wirawan, DEA.

## **Abstrak :**

Kebutuhan manusia semakin hari semakin kompleks. Tidak dapat dipungkiri, sejalan dengan pemenuhan kebutuhan manusia yang semakin besar itu, maka perkembangan teknologi secara otomatis akan mengikutinya. Sebagai contoh adalah dalam bidang teknologi kamera pengawas atau bisa disebut juga *Visual Sensor Network*. Namun teknologi ini masih terbelang masih dalam taraf berkembang, masih terdapat beberapa permasalahan didalamnya. Salah satu masalah yang menjadi kendala dalam perkembangan teknologi ini adalah keterbatasan sumber energi. Salah satu solusinya adalah dilakukan penghematan energi yang dikonsumsi tanpa mengurangi fungsi dan performa dari kamera tersebut. Dengan memanfaatkan kamera sebagai sensor pendeteksi pergerakan suatu objek yang melintas dalam daerah jangkauannya, maka kamera tersebut hanya melakukan pengambilan gambar ketika terdapat pergerakan dalam daerah jangkauannya.

Hasil dari penelitian ini adalah sistem *Visual Sensor Network* ini dapat menghemat daya sebesar  $\pm 19,6\%$  dan bit energi  $\pm 13,71\%$ . Selain itu, dengan algoritma ini dapat pula menghemat kapasitas memori karena menghasilkan output video yang ukurannya lebih kecil dengan resolusi yang sama.

**Kata kunci :** *visual sensor network*, konsumsi energi, node

# VISUAL SENSOR NETWORK WITH ENERGY SAVING ALGORITHM

Hananto Agung Baskoro  
2211 100 150

1<sup>st</sup> Advisor : Eko Setijadi, ST., MT., Ph.D

2<sup>nd</sup> Advisor : Dr. Ir. Wirawan, DEA.

## **Abstract :**

*In this globalization era, human needs becomes more complex and the increasing demand of human needs make technology must follow those demand. For example one of those technology is Visual Sensor Network. But these technology still in development state which still have various problem. One of the problem in Visual Sensor Network is the limited of energy source. So one of the methods to solve those problem is to decrease energy consumption while maintain camera's performancy and function. Using camera as the sensor to detect movement of object in camera's coverage, camera only record while there is a movement in its coverage.*

*The result shows that Visual Sensor Network reduce energy consumption  $\pm 19.6\%$ , energy bit  $\pm 13.71\%$  and memory because the video size is also reduce while maintain the resolution.*

**Keywords :** *visual sensor network, energy consumption, node*

## **KATA PENGANTAR**

Alhamdulillah, puji syukur penulis panjatkan kehadirat Allah SWT karena atas rahmat dan karunia-Nya penulis dapat menyelesaikan penelitian dan penulisan buku Tugas Akhir dengan judul :

### ***VISUAL SENSOR NETWORK DENGAN ALGORITMA YANG HEMAT ENERGI***

Tugas akhir merupakan salah satu syarat yang harus dipenuhi untuk menyelesaikan program studi Strata-1 pada Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Penulis menyadari bahwa dalam penulisan tugas akhir ini banyak mengalami kendala, namun berkat bantuan, bimbingan, dan kerjasama dari berbagai pihak sehingga kendala-kendala tersebut dapat diatasi. Rasa syukur kehadirat Allah SWT yang telah memberikan petunjuk, kemudahan dan kemurahan-Nya dalam mengatasi permasalahan selama proses penelitian ini. Serta tidak lupa penulis menyampaikan banyak terimakasih dan penghargaan setinggi-tingginya kepada :

1. Nabi Muhammad SAW, yang Insya Allah selalu menjadi panutan dalam setiap langkah hidup.
2. Kedua orang tua, yang selalu mendukung dari berbagai bidang dan selalu mengirimkan doa, dan menjadi motivasi utama penulis yang selalu menguatkan dalam menghadapi kesulitan dan keraguan.
3. Tete dan Fauz, yang selalu memberikan semangat, dukungan materi dan doa untuk penulis dalam penyelesaian penelitian ini.
4. Bapak Eko Setijadi ST., MT., Ph.D dan Dr. Ir. Wirawan, DEA selaku dosen pembimbing atas bimbingan, dukungan moral telah diberikan mulai dari awal penelitian ini hingga proses akhir penelitian.
5. Bapak dan Ibu dosen Teknik Elektro ITS, khususnya dosen Telekomunikasi Multimedia yang telah sangat berjasa memberikan ilmu kepada para mahasiswa.
6. Rekan yang telah membantu dalam penelitian ini, Andi Muhammad, Novem Ardan, Hanifar Kahira, dan Ilham Laenur.

7. Rekan seperjuangan tugas akhir Fadli Rahmawan, Hizrian Hizburrahman, dan Putri Nur Annisa Rufaidah yang telah menemani dan berbagi waktu.
8. Septian Hadi, M. Naufal Farid, Johanna Aprillia, Rizal Nur Ibrahim, Hidayah, Maria Yosefa dan M. Fanani yang telah membantu dalam pengambilan data dalam penelitian.
9. Rekan program master Mas Asriadi, Mas Dhany Riyanto, Mas Dukil, Mas Aristo, dan Bang Wira yang telah memberi saran dan masukan sangat berarti.
10. Teman-teman bidang studi TMM 2011 dan 2012 yang selalu kompak dan solid.
11. KONS58 keluarga penulis di Surabaya, Jiyi Nur Fauzan, Dzikri Aulia, Cahyo Septianto Hutomo, Leody Hazwendra, Virgasena Nabhan Ghaziyad, Dewanto Adi Prabowo, Ferry Muhammad, Yahyasalam Taslim.
12. Serta semua pihak yang telah membantu kelancaran pelaksanaan Tugas Akhir yang tidak dapat disebutkan keseluruhan.

Penulis menyadari bahwa pada penyusunan laporan tugas akhir ini masih terdapat kekurangan karena keterbatasan kemampuan yang penulis miliki, walaupun demikian penulis berharap tugas akhir ini dapat bermanfaat untuk kehidupan kedepannya.

Surabaya, 23 Juni 2015  
Penulis

## DAFTAR ISI

|   |      |
|---|------|
| COVER .....   | i    |
| PERNYATAAN KEASLIAN .....   | v    |
| HALAMAN PENGESAHAN .....  | vii  |
| ABSTRAK .....   | ix   |
| ABTRACT .....   | xi   |
| KATA PENGANTAR .....  | xiii |
| DAFTAR ISI .....  | xv   |
| DAFTAR GAMBAR .....   | xix  |
| DAFTAR TABEL .....  | xxi  |
| BAB 1 PENDAHULUAN .....   | 1    |
| 1.1 Latar Belakang .....  | 1    |
| 1.2 Perumusan Masalah .....   | 1    |
| 1.3 Batasan Masalah .....   | 2    |
| 1.4 Tujuan .....  | 2    |
| 1.5 Metodologi .....  | 2    |
| 1.6 Sistematika Pembahasan .....  | 3    |
| 1.7 Relevansi .....   | 3    |
| BAB 2 TEORI DASAR .....   | 5    |
| 2.1 <i>Wireless Sensor Network</i> .....  | 5    |
| 2.1.1 Gambaran Singkat <i>Wireless Sensor Network</i> .....                                 | 6    |
| 2.1.2 Tipe Aplikasi <i>Wireless Sensor Network</i> .....                                    | 8    |
| 2.1.3 Perkembangan <i>Wireless Sensor Network</i> .....                                     | 9    |
| 2.1.4 Tantangan Untuk <i>Wireless Sensor Network</i> s .....                                | 10   |
| 2.2 <i>Visual Sensor Networks</i> .....   | 12   |
| 2.2.1 Perbedaan <i>Wireless Sensor Network</i> s dengan <i>Visual Sensor Networks</i> ..... | 13   |
| 2.2.2 Contoh Pengimplemantasian <i>Visual Sensor Networks</i> .....                         | 14   |
| 2.2.3 Performansi Optimal <i>Visual Sensor Networks</i> dengan Mikroprosesor .....          | 14   |



|   |    |
|---|----|
| 2.2.4 Modus Operasi pada <i>Visual Sensor Networks</i> .....        | 15 |
| 2.4 Pemrograman pada Video .....                                    | 15 |
| 2.4.1 Pengertian Algoritma .....                                    | 16 |
| 2.4.2 Teknik <i>Background Subtraction</i> .....                    | 16 |
| 2.4.3 Teknik Kompresi Video .....                                   | 17 |
| 2.4.4 Teknik <i>Multithread</i> .....                               | 18 |
| 2.4.5 Teknik <i>Socket Communication</i> .....                      | 20 |
| 2.5 Mikroprosesor .....   | 21 |
| 2.5.1 Karakteristik Mikroprosesor .....                             | 22 |
| 2.5.2 Komponen Dasar Sistem Mikroprosesor .....                     | 22 |
| 2.6 Raspberry Pi .....  | 23 |
| 2.6.1 Spesifikasi Raspberry Pi B+.....                              | 24 |
| 2.6.2 Desain Raspberry Pi Model B .....                             | 25 |
| 2.7 Konsep Citra dan Video .....                                    | 25 |
| 2.7.1 Konsep Citra .....  | 25 |
| 2.7.3 Konsep Video.....   | 26 |
| BAB 3 PERANCANGAN DAN IMPLEMENTASI .....                            | 29 |
| 3.1 Perancangan Uji Unjuk Kerja Sistem.....                         | 30 |
| 3.2 Kebutuhan Pendukung Infrastruktur .....                         | 31 |
| 3.2.1 Perangkat keras (Hardware).....                               | 31 |
| 3.2.2 Perangkat Lunak ( <i>Software</i> ).....                      | 33 |
| 3.3 Instalasi Perangkat Lunak (Software).....                       | 35 |
| 3.3.1 Instalasi Operational System Raspberry Pi .....               | 35 |
| 3.4 Desain Sistem Kerja .....                                       | 37 |
| 3.4.1 Skema dan Sistem Kerja <i>Visual Sensor Networks</i> .....    | 37 |
| 3.4.2 Lokasi Pengimplementasian <i>Visual Sensor Networks</i> ..... | 38 |
| 3.5 Algoritma Implementasi <i>Visual Sensor Networks</i> .....      | 40 |
| 3.5.1 Simulasi Kerja Sederhana .....                                | 41 |

|  |    |
|--|----|
| 3.5.2 Teknik <i>Background Subtraction</i> .....                       | 42 |
| 3.5.3 Teknik Kompresi .....  | 44 |
| 3.5.4 Teknik Socket.....   | 45 |
| 3.5.5 Teknik Multithread .....   | 47 |
| 3.6 Implementasi Sistem <i>Visual Sensor Networks</i> .....            | 49 |
| 3.6.1 Pemanggilan Program pada <i>Visual Sensor Networks</i> .....     | 49 |
| 3.6.2 Komunikasi Server-Client Pada Raspberry Pi.....                  | 50 |
| 3.6.3 Pendeteksian Objek dan Metode Kompresi Resolusi .....            | 51 |
| 3.6.3 Penyimpanan Data Citra Hasil Uji Coba .....                      | 52 |
| 3.7 Skenario Pengujian .....   | 52 |
| 3.7.1 Blok Diagram Pengujian .....                                     | 52 |
| 3.7.1 Parameter Pengujian.....   | 53 |
| 3.8 Pengambilan Data .....   | 53 |
| 3.8.1 Metode Pengambilan Data .....                                    | 53 |
| 3.8.2 Pengolahan Data.....   | 54 |
| BAB 4 ANALISIS DATA .....  | 57 |
| 4.1 Analisis Algoritma <i>Visual Sensor Networks</i> .....             | 57 |
| 4.1.1 Skema Ruang dan Daerah <i>Overlap</i> .....                      | 57 |
| 4.1.2 Analisis Sistem Simulasi Sederhana.....                          | 58 |
| 4.1.3 Analisis Sistem Implementasi <i>Visual Sensor Networks</i> ..... | 63 |
| 4.2 Analisis Pengaruh Konsumsi Daya.....                               | 66 |
| 4.2.1 Daya Tanpa Penggunaan Algoritma.....                             | 67 |
| 4.2.2 Daya dengan Algoritma.....                                       | 70 |
| 4.2.3 Perbandingan Daya.....   | 74 |
| 4.3 Analisis Energi Bit Rate .....                                     | 74 |
| 4.3.1 Analisis Ukuran Video .....                                      | 75 |
| 4.3.2 Analisa Perhitungan Bit Rate .....                               | 76 |
| 4.3.3 Analisa Kebutuhan Bit Energi .....                               | 77 |

|   |     |
|---|-----|
| BAB 5 KESIMPULAN DAN SARAN .....  | 81  |
| 5.1 Kesimpulan.....   | 81  |
| 5.2 Saran.....  | 81  |
| DAFTAR PUSTAKA .....  | 83  |
| BIOGRAFI PENULIS .....  | 84  |
| LAMPIRAN .....  | 87  |
| A. Instalasi Matlab R2011b .....  | 87  |
| B. Instalasi OpenCV .....   | 89  |
| C. Instalasi C++ .....  | 89  |
| D. Program Simulasi Sederhana dengan Menggunakan Matlab<br>R2011b ..... | 91  |
| E. Program Node Server Tanpa Menggunakan Algoritma .....                | 92  |
| F. Program Node Client Tanpa Menggunakan Algoritma.....                 | 96  |
| G. Program Node Server dengan Menggunakan Algoritma .....               | 99  |
| H. Program Node Client dengan Menggunakan Algoritma .....               | 102 |

## DAFTAR TABEL

|   |    |
|---|----|
| Tabel 3.1 Tabel Pengelompokkan Resolusi dan Frame Rate .....              | 44 |
| Tabel 3.2 Tabel Perintah Data untuk Socket Communication .....            | 47 |
| Tabel 3.3 Perubahan IP dan Subnet Mask Pada Raspberry Pi .....            | 50 |
| Tabel 4.1 Tabel Pengukuran Data Arus Tanpa Algoritma .....                | 69 |
| Tabel 4.2 Tabel Pengukuran Data Tegangan Tanpa Algoritma .....            | 69 |
| Tabel 4.3 Tabel Pengukuran Data Daya Tanpa Algoritma .....                | 70 |
| Tabel 4.4 Tabel Pengukuran Data Arus dengan Algoritma .....               | 73 |
| Tabel 4.5 Tabel Pengukuran Data Tegangan dengan Algoritma.....            | 73 |
| Tabel 4.6 Tabel Pengukuran Data Daya dengan Algoritma.....                | 73 |
| Tabel 4.7 Tabel Perbandingan Daya Rata-rata.....                          | 74 |
| Tabel 4.8 Tabel Data Frame Rate Video.....                                | 75 |
| Tabel 4.9 Perbandingan Hasil Pengukuran dan Perhitungan Ukuran File ..... | 75 |
| Tabel 4.10 Perbandingan Bit Rate Kedua Metode Pengambilan Gambar .....    | 77 |
| Tabel 4.11 Perbandingan Bit Energi .....                                  | 78 |

## DAFTAR GAMBAR

|   |    |
|---|----|
| Gambar 2.1 Contoh Pemodelan Sistem <i>Wireless Sensor Network</i> ..... | 6  |
| Gambar 2.2 <i>Wireless Sensor Network</i> s dalam IoT .....             | 10 |
| Gambar 2.3 Contoh Skenario <i>Visual Sensor Networks</i> .....          | 13 |
| Gambar 2.4 Contoh Event Algoritma Background Substraction .....         | 17 |
| Gambar 2.5 Proses Kompresi Data Secara Umum.....                        | 18 |
| Gambar 2.6 Proses Kompresi Data Secara Lossless dan Lossy .....         | 18 |
| Gambar 2.7 Ilustrasi Single dan Multi Thread .....                      | 19 |
| Gambar 2.8 Model Sistem Kerja Client-Server .....                       | 21 |
| Gambar 2.9 Raspberry Pi B+ .....  | 24 |
| Gambar 3.1 Diagram Alur Penelitian.....                                 | 29 |
| Gambar 3.2 Webcam M-Tech 1.3 Megapixel .....                            | 32 |
| Gambar 3.3 Tampilan Desktop Raspberry Pi.....                           | 36 |
| Gambar 3.4 Tampilan Booting Raspberry Pi .....                          | 37 |
| Gambar 3.5 Blok Diagram Sistem .....                                    | 53 |
| Gambar 4.1 Daerah Jangkauan Node Client .....                           | 58 |
| Gambar 4.2 Daerah Jangkauan Node Server.....                            | 58 |
| Gambar 4.3 Frame Referensi Ruang Kosong .....                           | 59 |
| Gambar 4.4 Frame Referensi dengan Objek .....                           | 59 |
| Gambar 4.5 Hasil Output <i>Grayscale</i> .....                          | 60 |
| Gambar 4.6 Hasil Citra Setelah Proses <i>Erode Image</i> .....          | 61 |
| Gambar 4.7 Hasil Citra Setelah Proses <i>Dilate Image</i> .....         | 61 |
| Gambar 4.8 Deteksi Pergerakan Objek Pada Simulasi Sederhana .....       | 62 |
| Gambar 4.9 Deteksi Koordinat Pixel .....                                | 62 |
| Gambar 4.10 Gambar Video Tanpa Algoritma Node Server .....              | 64 |
| Gambar 4.11 Gambar Video Tanpa Algoritma Node Client.....               | 64 |
| Gambar 4.12 Gambar Video dengan Algoritma Node Server.....              | 65 |
| Gambar 4.13 Gambar Video dengan Algoritma Node Client .....             | 65 |
| Gambar 4.14 Grafik Tegangan Node Server Tanpa Algoritma.....            | 67 |
| Gambar 4.15 Grafik Arus Node Server Tanpa Algoritma .....               | 68 |
| Gambar 4.16 Grafik Tegangan Node Client Tanpa Algoritma .....           | 68 |
| Gambar 4.17 Grafik Arus Node Client Tanpa Algoritma .....               | 69 |
| Gambar 4.18 Grafik Tegangan Node Server dengan Algoritma .....          | 71 |
| Gambar 4.19 Grafik Arus Node Server dengan Algoritma .....              | 71 |
| Gambar 4.20 Grafik Tegangan Node Client dengan Algoritma.....           | 72 |
| Gambar 4.21 Grafik Arus Node Client dengan Algoritma.....               | 72 |

## **BAB 1**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Perkembangan yang cepat pada bidang jaringan sensor nirkabel telah memacu peneliti untuk bekerja membangun sensor yang memiliki *physical* yang lengkap. Jaringan yang seperti itu dinamakan sebagai *Wireless Sensor Network* (WSN). Salah satu dari aplikasi yang potensial dari WSN adalah berupa *Visual Sensor Network* (VSN). VSN mendeteksi citra, baik citra gambar maupun citra video. Setiap benda yang dipantau terhubung dengan node di WSN. Hal ini secara khusus penting jika benda yang dipantau besar dan di luar jangkauan. Namun aplikasi seperti ini membutuhkan konsumsi energi yang besar untuk citra gambar dan citra video sesuai dengan resolusinya. Pengambilan gambar pada VSN ini berada pada cakupan daerah tertentu. Proses penyebaran data harus efisien dikarenakan energi, pemrosesan, dan memori membebani node WSN, dan biasanya catu daya dari node di WSN berasal dari baterai yang memiliki daya terbatas dan juga sulit jika harus sering dilakukan pengantian di lingkungan yang dideskripsikan. Maka diperlukan metode untuk mengurangi pemakaian daya sebaik mungkin dan seminimal mungkin, namun tetap menghasilkan resolusi yang baik dan stabil. Salah satu faktor terbesar pemakaian daya adalah proses korelasi dari sensor tersebut. Semakin sering sensor berkorelasi maka semakin banyak pula daya yang terpakai.

Tugas akhir ini menawarkan solusi untuk masalah tersebut dengan menggunakan uji coba terhadap VSN dengan memaksimalkan algoritma yang hemat energi. Sehingga dengan resolusi yang tinggi sekalipun, apabila VSN mengaplikasikan algoritma yang hemat energi, maka konsumsi energi dari VSN tersebut akan terbilang rendah tetapi resolusi dari citra yang berhasil ditangkap pada jangkauan daerah tertentu akan tetap memiliki resolusi yang baik dan stabil.

#### **1.2 Perumusan Masalah**

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Bagaimana membuat simulasi algoritma dari *Visual Sensor Network* yang hemat energi?
2. Bagaimana pengaruh konsumsi energi yang rendah bila dibandingkan dengan yang menggunakan energi tinggi?
3. Bagaimana algoritma yang dipakai bekerja?

### 1.3 Batasan Masalah

Hal-hal yang akan dilakukan dalam penelitian ini adalah:

1. Perancangan *Visual Sensor Network* yang hemat energi
2. Pengujian akan dilakukan 2 kali, yaitu simulasi dengan perangkat lunak dan implementasi platform dengan alat yang ada di laboratorium
3. Citra yang akan diujikan pada *Visual Sensor Network* ini adalah citra gambar dengan resolusi sedang sampai tinggi

### 1.4 Tujuan

Tujuan dari tugas akhir ini adalah:

1. Merancang simulasi yang digunakan menggunakan aplikasi MATLAB
2. Mengimplementasikan metode yang digunakan dengan menggunakan perangkat keras yang dibutuhkan
3. Mendapatkan perbandingan rata-rata jumlah energi yang dapat dihemat menggunakan metode ini

### 1.5 Metodologi

Metode penelitian dilakukan dengan lima tahap yaitu studi literatur, perancangan simulasi sistem kerja VSN, pengujian dan pengukuran implementasi platform, pengolahan data hasil pengukuran, analisis data dan penarikan kesimpulan dari penelitian

#### 1. Studi literatur

Studi literatur dilakukan dengan mencari dan mempelajari beberapa paper dan jurnal baik skala nasional maupun internasional yang dapat menunjang pengerjaan tugas akhir ini. Pada tahap ini akan dipelajari cara kerja dari sistem jaringan WSN tersebut khususnya VSN.

#### 2. Perancangan Simulasi Sistem Kerja *Visual Sensor Network*

Pada tahap ini akan diterapkan simulasi berupa uji coba sistem kerja secara perangkat lunak. Pengerjaan simulasi menggunakan perangkat lunak Matlab 2011. Hal ini merupakan pengujian tahap awal untuk mendapatkan hasil yang direncanakan.

#### 3. Pengujian dan Pengukuran Implementasi Platform

Pada tahap ini dilakukan pengujian perangkat keras. Penelitian ini menggunakan kamera webcam M-Tech dan mikroprosesor Raspberry Pi seri B+.

#### **4. Pengolahan Data Hasil Pengukuran**

Pada tahap ini dilakukan pengolahan data berdasarkan hasil yang didapat dari uji coba simulasi menggunakan Matlab, maupun pengukuran menggunakan kamera dan mikroprosesor.

#### **5. Analisis Data dan Penarikan Kesimpulan**

Pada tahap ini dilakukan pengamatan dan analisa terhadap data yang telah diperoleh, beserta penarikan kesimpulan berdasarkan analisa data yang telah dilakukan.

### **1.6 Sistematika Pembahasan**

Laporan tugas akhir ini terdiri dari lima bab dengan sistematika penulisan sebagai berikut.

#### **- BAB 1 PENDAHULUAN**

Pada bab ini akan diuraikan mengenai latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

#### **- BAB 2 TEORI DASAR**

Pada bab ini akan dijelaskan tentang tinjauan pustaka yang akan membahas tentang konsepsi VSN, karakteristik VSN, mikroprosesor Raspberry Pi, teknik pencitraan, dan parameter unjuk kerja

#### **- BAB 3 PERANCANGAN DAN IMPLEMENTASI**

Pada bab ini akan dijelaskan tentang pengimplementasian sistem berdasarkan teori pada Bab 2, serta melakukan pengujian unjuk kerja sistem.

#### **- BAB 4 ANALISIS DATA**

Pada bab ini akan ditampilkan hasil pengujian, kemudian dilakukan analisa dari data yang telah diperoleh sehingga dapat memudahkan melakukan penarikan kesimpulan.

#### **- BAB 5 KESIMPULAN DAN SARAN**

Pada bab ini berisi tentang kesimpulan dan saran berdasarkan pengambilan data dari implementasi yang telah dilakukan dalam penelitian tugas akhir ini.

### **1.7 Relevansi**

Hasil yang didapat dari tugas akhir ini diharapkan dapat memberi manfaat sebagai berikut :

1. Dapat memberikan kontribusi berupa algoritma yang bisa menghemat energi pada pemakaian sistem kerja *Visual Sensor*



*Network* dan dapat membantu menemukan solusi energi dalam penerapan *Visual Sensor Network*.

2. Menjadi referensi dalam pengimplementasian *Visual Sensor Network* yang lebih efisien dan berdaya rendah.

## **BAB 2**

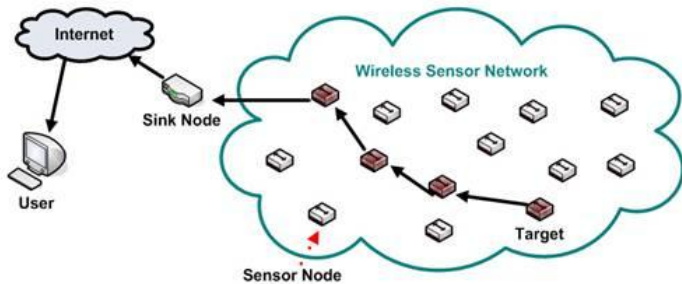
### **TEORI DASAR**

#### **2.1 *Wireless Sensor Network* [2]**

Penginderaan manusia adalah salah satu cara untuk mengumpulkan informasi, memperoleh pengetahuan, dan membuat keputusan yang dapat diandalkan. Jaringan Sensor Nirkabel (JSN) atau *Wireless Sensor Network* (WSN) meniru kemampuan kecerdasan manusia ini, tetapi pada skala yang lebih luas, dengan lebih cepat, lebih murah, dan cara yang lebih efektif sehingga dapat digunakan untuk berbagai aplikasi.

WSN adalah sebuah jaringan yang tersusun atas berbagai sensor node yang memiliki kemampuan penginderaan, komunikasi secara nirkabel dan juga komputasi merupakan infrastruktur suatu jaringan yang terdiri dari sekumpulan node sensor yang tersebar pada suatu area sensor. Tiap node sensor memiliki kemampuan untuk mengumpulkan data disekitarnya dan meroutingkan kembali ke sink node melalui transmisi radio secara intensif. Data yang dikumpulkan bisa berupa suhu, tekanan, pergerakan suatu objek atau kelembaban dan lain sebagainya.

Pada umumnya JSN atau WSN terdiri dari dua komponen, yaitu node sensor dan sink. Node sensor merupakan komponen kesatuan dari jejaring yang dapat menghasilkan informasi, biasanya merupakan sebuah sensor atau juga dapat berupa sebuah aktuator yang menghasilkan *feedback* pada keseluruhan operasi. Sensor disebar dengan volume dan kerapatan yang tinggi. Sink merupakan kesatuan proses pengumpulan informasi dari node sensor sehingga dapat dilakukan pengolahan informasi lebih lanjut. Adanya karakteristik tersebut, perlu adanya metodologi yang mampu melampaui karakteristik-karakteristik pada WSN serta tidak membatasi pengiriman informasi, network, manajemen operasional, kerahasiaan, integritas, dan proses di dalam network. Aplikasi teknologi WSN memungkinkan peneliti untuk mendapat informasi yang maksimal tanpa harus berada di area sensor. Informasi dapat diakses dari jarak jauh melalui gadget seperti laptop, remote control, server dan sebagainya.



**Gambar 2.1 Contoh Pemodelan Sistem *Wireless Sensor Network***

Sensor-sensor pada WSN akan mengubah data analog ke data digital. Data hasil pengubahan ini akan dikirim ke suatu node atau node lainnya melalui beberapa media komunikasi yaitu:

- Bluetooth
- Infrared
- Wi-fi
- Kabel USB, dan lainnya

Kemampuan sensor pada WSN secara luas membuat penggunaannya untuk melakukan kegiatan monitoring banyak digunakan. WSN dapat digunakan dengan sensor sederhana yang memonitor suatu fenomena atau suatu kejadian. Pada pengaplikasian yang kompleks, setiap WSN akan mempunyai lebih dari satu sensor, sehingga WSN ini akan banyak melakukan monitoring suatu kejadian atau fenomena dalam satu kali kejadian. WSN juga memiliki keunggulan lainnya yaitu dapat diintegrasikan ke *gateway* yang dapat mengakses internet, maka WSN dapat diakses dengan perangkat yang terhubung internet.

### **2.1.1 Gambaran Singkat *Wireless Sensor Network***

Perkembangan dan pengembangan yang pesat dari teknologi terbaru ini sangat bergantung sekali dengan perkembangan dan pengembangan perangkat keras (*hardware*) seperti semi-konduktor, sistem pengembangan perangkat lunak (*software*), kebijakan individu, sumber energi (*battery*), dan efisiensi energi. Perbedaan aplikasi juga

turut menentukan perbedaan kebutuhan sensor dan topologi jaringan sensor. Kemampuan dari jangkauan (*coverage*) sangat beragam tergantung pada tujuan dari sistem yang dibangun. Teknik dan metode pengukuran juga turut mempengaruhi dan sangat beragam dari satu aplikasi ke aplikasi lainnya. Pengembangan dari teknologi ini terus dilakukan, dengan tujuan nantinya lingkungan dan objek kerja yang belum terjangkau bisa dengan perlahan diterapkan teknologi monitoring WSN ini. Walaupun teknologi ini terus mengalami diversitas secara operasional dan tujuan, akan tetapi ada beberapa poin penting yang saling bersangkutan untuk dapat membangun jaringan sensor yang optimal, antara lain:

- *Low Power*

Pemantauan terhadap suatu kejadian dan lingkungan sangat tidak memungkinkan untuk terus berhubungan dengan suatu catu daya yang selalu tersedia. Kebutuhan akan sumber tenaga menjadi target utama suatu sistem yang rendah akan konsumsi tenaga. Penggunaan baterai sebagai sumber tenaga merupakan suatu solusi populer dan cukup kuat dalam pemantauan jangka panjang.

- *Low Cost*

Ketika jangkauan area daripada target melebar, serta dibutuhkan tingkat resolusi yang tinggi, dengan kejadian seperti itu dibutuhkan node yang lebih banyak atau lebih tinggi kualitasnya. Apabila tetap berusaha untuk memenuhi keadaan tersebut, tetapi dibatasi dengan keterbatasan sumber tenaga, dan lingkungan pemantauan yang kurang bersahabat, maka dibutuhkan minimal beberapa jaringan yang terdiri dari penyebaran sensor dengan pengaturan otomatis.

- *Self Organizing*

Apabila sebuah aplikasi sederhana diatasi oleh jaringan sensor dengan jumlah yang cukup banyak dan bahkan sangat banyak, maka menyebabkan kesulitan bagi operator untuk dapat mengoperasikannya. Keadaan alam yang dinamis mengharuskan sensor memiliki kemampuan pengaturan. Apabila kriteria ini terpenuhi, maka bentuk jaringan *adhoc* merupakan pilihan utama dalam jaringan sensor nirkabel. Setiap sensor yang dapat mengorganisir diri, memungkinkan mereka merutekan data dan pesan secara optimal. Layanan ini *location aware* sangat diperlukan pula dalam melakukan agresi suatu data. Saat mendesain sistem WSN untuk kondisi suatu kasus, perlu dipertimbangkan pula hal-hal

seperti *power constrain*, maupun pertimbangan lainnya agar sistem dapat bekerja efisien dan optimal.

### 2.1.2 Tipe Aplikasi *Wireless Sensor Network*

Kebanyakan tipe aplikasi dari WSN dibawah ini memiliki karakteristik dasar yang sama. Terdapat perbedaan yang sangat jelas, yaitu pada sumber data (hasil pembacaan sensor), dan sink (suatu titik dimana data pembacaan tersebut harus dikirimkan). Sink juga sering dianggap sebagai sistem jaringan sensor itu sendiri, tetapi dalam beberapa contoh kasus monitoring, sink tidak terlalu dibutuhkan, karena yang terpenting adalah data hasil pembacaan monitoring. Contoh interaksi antara sensor dan sink ditunjukkan pada beberapa tipe aplikasi, dan beberapa yang paling relevan, antara lain:

- **Deteksi Kejadian (*Event Detection*)**  
Pada tipe aplikasi ini setiap sensor node harus melakukan pengiriman laporan (*report*) kepada sink, tepat setelah sensor tersebut mendeteksi munculnya suatu kejadian. Kejadian yang tergolong sederhana, dapat dioperasikan dengan menggunakan sensor tunggal, sedangkan untuk kejadian yang lebih kompleks dibutuhkan kolaborasi dari beberapa node sensor sehingga dapat dilakukan pengambilan keputusan apakah suatu kejadian sering terjadi. Apabila muncul beberapa kejadian, maka pengklarifikasian kejadian dapat menjadi solusi yang tepat.
- **Pengukuran Berkala (*Periodic Measurement*)**  
Sensor dapat diberi tugas untuk melakukan reportase dari hasil pengukuran secara periodik. Kondisi pengecualian sistem ini tetap dapat melakukan report sesegera mungkin setelah suatu kejadian muncul (*triggered report*)
- **Pendekatan fungsi dan deteksi sisi terluar (*Function approximation and edge detection*)**  
Perilaku ketika suatu nilai fisik berubah dari suatu tempat ke tempat lainnya dapat direpresentasikan sebagai fungsi posisi. Sebuah jaringan sensor nirkabel dapat diterapkan dengan cara mengambil beberapa sampel data dari sensor node. Pendekatan ini harus tersedia di sink dalam bentuk *mapping*. Bagaimana dan kapan saatnya *update* pada *mapping* bergantung pada kebutuhan aplikasinya. Cara yang sama dilakukan untuk mendeteksi nilai sisi terluar. Salah satu contoh aplikasinya adalah mencari sisi terluar dari kebarakaran hutan, maka

dapat dicari dengan mencari sisi (*edge*) dari fungsi tersebut dengan cara mengirim pesan ke titik sensor terluar.

- Pencarian jejak (*tracking*)

Sumber dari suatu kejadian dapat berupa benda yang bergerak. WSN dapat digunakan untuk melaporkan posisi dari sumber sink dan dilengkapi pula dengan arah dan kecepatan dari sumber kejadian tersebut. Hal yang harus dilakukan untuk mencapai hal itu adalah masing-masing sensor node harus sudah terintegrasi sebelum data terbaru dikirim ke sink.

### 2.1.3 Perkembangan *Wireless Sensor Network*

Pada awalnya, penelitian mengenai WSN adalah merupakan bagian dari program *Distributed Sensor Network* (DSN) oleh *Defense Advanced Research Projects Agency* (DARPA) sekitar tahun 1980. Sensor pada masa itu sangat mahal dan memiliki ukuran relatif besar sehingga penggunaannya sangat terbatas, terutama untuk militer atau ketahanan negara. Pada perkembangannya saat ini banyak kita jumpai sensor-sensor berukuran yang relatif kecil dengan harga yang terjangkau sehingga sensor dapat digunakan dalam aplikasi yang lebih luas sejalan dengan perkembangan teknologi internet. WSN kemudian terintegrasi dalam konsep Internet of Things (IoT), biasa juga disebut IP-based WSN. Pada konsep ini, setiap sensor memiliki identitas/ alamat yang unik sehingga mampu secara dinamis bergabung dalam suatu jaringan, berkolaborasi dan bekerja sama secara efisien. Beberapa dari pengembangan aplikasi dari WSN antara lain:

- *Smart Home*

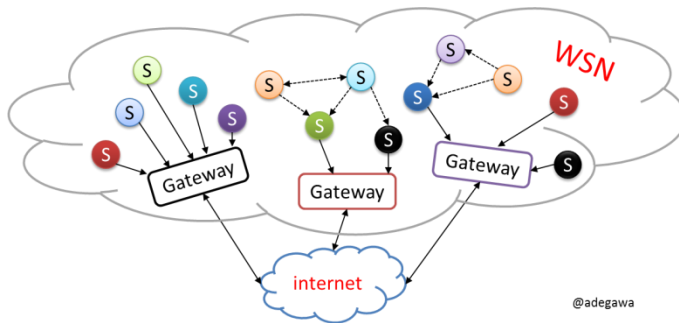
Sensor-sensor pada WSN mampu mengukur berbagai macam data lingkungan. Misalnya suhu tingkat tinggi/ rendah suara, intensitas cahaya, dan lain sebagainya, sehingga *device* atau dalam hal ini merupakan manusia dapat mengambil keputusan berdasarkan perubahan yang terjadi pada lingkungan sekitar rumah. Salah satu contohnya adalah perubahan suhu udara di luar ruangan dapat diketahui dengan perubahan warna lampu dalam ruangan. Pengembangan lainnya adalah dalam pengaturan sistem pemanas ruangan berdasarkan keberadaan manusia di dalam ruangan atau berdasarkan cuaca di luar.

- *Driverless Vehicles/Automobiles*

Sensor-sensor yang terpasang pada beberapa bagian suatu mobil memberikan informasi yang berguna bagi pusat pemrosesan data di mobil tersebut untuk mencegah terjadinya tabrakan.

- *Wearable Healthcare Devices*

Perkembangan kesehatan tubuh kita dapat dimonitor dengan menggunakan sensor yang ditanamkan dalam pakaian, dan masih banyak lagi aplikasi pengembangan dari teknologi terkini WSN.



**Gambar 2.2** *Wireless Sensor Networks* dalam IoT

Teknologi WSN semakin populer, dan kelak dengan kemajuan teknologi WSN dan didukung oleh infrastruktur IoT maka akan menciptakan suatu sistem saling terintegrasi yang dapat membentuk sistem WSN berbasis IoT. Gambar diatas, tiap node S digambarkan sebagai node dari sistem *sensing* WSN yang terhubung pada masing-masing *gateway* dan terhubung secara langsung kepada internet.

#### 2.1.4 Tantangan Untuk *Wireless Sensor Networks*

Kebutuhan untuk menangani aplikasi dengan jangkauan yang luas mempersulit kemungkinan realisasi dari jaringan sensor nirkabel. Oleh karena itu beberapa sifat dan karakter yang umum bermunculan, terutama yang berkaitan dengan karakter mekanisme. Contoh karakteristik yang menjadi syarat dari kebanyakan aplikasi WSN antara lain:

- Tipe layanan (*Type of Service*)

Pada WSN data yang ditransmisikan mulai dari sensor terluar sampai enduser hanya berbentuk bit saja. Hal ini tentu saja bukan tujuan

utama dari teknologi ini, dimana yang diharapkan dari teknologi ini adalah pemberian informasi yang dapat dimengerti tentang suatu kejadian. Oleh karena itu paradigma baru tentang penggunaan konsep jaringan dalam sistem sensor merubah cara berpikir orang mengenai layanan apa yang bisa diberikan oleh suatu jaringan.

- Kualitas Layanan QoS (*Quality of Service*)

Kebutuhan akan QoS secara tradisional biasanya muncul dari aplikasi yang bertipe multimedia. Permasalahan yang sering muncul seperti *delay* dan *bandwidth* minimum tidak relevan ketika suatu aplikasi sangat membutuhkan *latency* saat data akan dikirimkan. Pada beberapa kasus terkadang hanya memiliki kemampuan mengirimkan data, sedangkan pada kasus lainnya sangat dibutuhkan keandalan atau reability yang sangat tinggi. Pada kasus tertentu terkadang *delay* juga dibutuhkan apabila sebuah aktuatur diatur secara *real-time* oleh suatu jaringan sensor.

- Toleransi (*Fault Tolerance*)

Habisnya tenaga dari suatu node sensor, kerusakan dari node sensor, ataupun adanya interupsi permanen dari komunikasi nirkabel, mengharuskan suatu WSN untuk memiliki toleransi terhadap kesalahan. Apabila suatu sistem dapat mentolerir terhadap suatu kesalahan, maka proses *deploy* akan lebih mudah dilakukan karena pada masing-masing bagian daripada jaringan sudah berfungsi dengan benar.

- Waktu Kerja (*Lifetime*)

Pada suatu skenario suatu sensor harus tetap bekerja pada keadaan energi yang terbatas (menggunakan baterai), dan proses mengganti tenaga dengan yang baru sama sekali tidak praktis dan tidak dapat dilakukan secara simultan, hal ini mengingat WSN harus dapat bekerja sesuai sistem kerjanya selama mungkin. Dan semenjak faktor energi merupakan salah satu bagian yang krusial dan sangat penting untuk menunjang kinerja dari sistem kerja WSN, maka mulai banyak riset dan penelitian untuk merancang sistem kerja baru yang mengkonsumsi energi secara lebih efisien.

- Skalabilitas (*Scalability*)

Pada aplikasinya WSN sering menggunakan node dalam jumlah yang sangat besar, maka dalam perancangan arsitekturnya diperlukan perancangan dengan skala tertentu yang menunjang daripada kinerja keseluruhan sistem kerja.



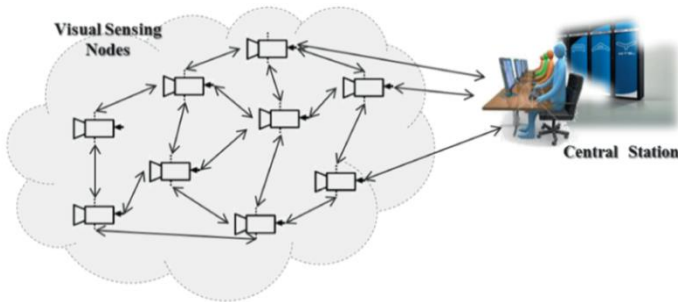
- **Kepadatan (*Sensor Density*)**  
Jumlah node per satu unit area sangat dipertimbangkan. Perbedaan aplikasi akan mempengaruhi pula kepadatan node yang memonitor suatu area pengawasan. Nilai kepadatan akan bervariasi dari waktu maupun ruang karena suatu node bisa saja mengalami kegagalan ataupun mengalami perpidahan
- **Kemampuan menerima program (*Programmability*)**  
Tidak hanya kemampuan untuk memproses informasi data, tetapi suatu WSN juga harus lebih fleksibel terhadap berbagai macam monitoring sistem. Oleh karena itu penting bagi tiap node untuk dapat bekerja sesuai program yang ditanamkan, dan program yang ditanamkan harus bisa dimodifikasi apabila harus dibutuhkan variasi pengamatan.
- ***Maintability***  
Baik lingkungan sebagai objek pengamatan maupun topologi dari suatu WSN mengalami perubahan, maka suatu sistem harus dapat dengan cepat beradaptasi dengan lingkungan barunya. Suatu sistem optimal harus bisa melakukan pengawasan kondisi status dari dirinya sendiri. Oleh karena itu perlu dibangun sistem yang dapat melakukan pemeliharaan terhadap dirinya sendiri (*self-maintenance*), atau juga dapat menerima pemeliharaan dari luar (*external maintenance*), sehingga sistem dapat beroperasi sesuai dengan kualitas yang diinginkan.

## 2.2 Visual Sensor Networks [1]

WSN hemat energi menjadi fokus dan penelitian aktif daerah di seluruh dunia. Perkembangan dari WSN yang sangat pesat dan kebutuhan akan perkembangan teknologi citra menyebabkan munculnya kelas baru jaringan yang didistribusikan berbasis sensor. Maka munculah beberapa pengembangan implementasi aplikatif lainnya. *Visual Sensor Networks* (VSN) adalah sebuah jaringan sensor visual dimana terdiri dari *smart camera* yang terdistribusi dalam suatu wilayah spasial yang mampu memproses dan menangkap gambar atau citra dari berbagai sudut pandang kedalam bentuk yang lebih kompleks dan sempurna dalam pengambilan citranya.

VSN merupakan jenis dari WSN, dan banyak teori serta aplikasi yang masih berkaitan antara keduanya. Sebuah jaringan pada

VSN umumnya terdiri dari beberapa kamera, yang memiliki beberapa pengolahan citra, komunikasi dan kemampuan penyimpanan lokal, serta satu atau lebih komputer pusat. Data gambar dari beberapa kamera diproses lebih lanjut dan dikirimkan kepada server. Ada beberapa aplikasi VSN yang biasanya memberikan beberapa layanan tingkat tinggi kepada pengguna sehingga sejumlah besar data dapat diproses menjadi informasi yang menarik menggunakan *query*. tertentu. Gambar 2.3 merupakan contoh skema node dari VSN. Pada gambar tersebut tiap node antar kamera saling berkomunikasi dengan dan ada suatu pusat server yaitu *central station*.



**Gambar 2.3 Contoh Skenario *Visual Sensor Networks***

### **2.2.1 Perbedaan *Wireless Sensor Networks* dengan *Visual Sensor Networks***

Hal utama yang membedakan antara VSN dengan jaringan sensor lainnya adalah terletak pada sifat dan volume informasinya. Tidak seperti kebanyakan sensor, kamera sebagai sensor yang mencakup suatu daerah spasial yang terjangkau, dan menangkap sejumlah besar data citra visual kemudian diproses dalam skema jaringan tertentu. Hal ini menjadi pembeda antara VSN dengan jaringan sensor lainnya. Ketika kebanyakan WSN digunakan sebagai alat pengambil data untuk pendeteksi kelembapan, temperatur, tekanan, kecepatan debit air, dan lain sebagainya. VSN memproses data visual. Selain itu perbedaan yang dapat kita lihat adalah antara WSN dan VSN adalah jenis data yang dikumpulkan dan diproses. Seperti yang dijelaskan diatas, bahwa WSN biasanya hanya memproses data skalar dari beberapa implementasi.

Pada hal ini komunikasi dalam jaringan sensor visual yang berbeda secara substansial dari jaringan sensor lainnya.

### **2.2.2 Contoh Pengimplemantasian *Visual Sensor Networks***

Aplikasi VSN banyak digunakan untuk mempermudah kerja manusia, khususnya dalam bidang pemantauan. Beberapa contoh aplikasi VSN adalah:

- Pengawasan lokasi publik dan komersial: VSN dapat digunakan untuk pemantauan publik pada tempat-tempat seperti taman, pusat perbelanjaan, sistem transportasi, dan lokasi produksi untuk memonitoring kerusakan infrastruktur, pendeteksi kecelakaan, dan pencegahan kejahatan
- Pemantauan lingkungan dan monitoring bangunan: VSN adalah solusi sempurna untuk pendeteksian bencana longsor, kebakaran, atau kerusakan di pantai dan gunung, lokasi situs bersejarah dan situs arkeologi dan beberapa lokasi pelestarian budaya daerah-daerah lainnya.
- Pengawasan militer: jaringan tersebut dapat digunakan dalam patroli petugas keamanan di daerah perbatasan yang rawan konflik, mengukur arus pengungsi daerah konflik, dan membantu mengontrol *battlefield command* dalam medan tempur.
- *Smart homes and meeting rooms*: VSN dapat memberikan pemantauan secara terus menerus terhadap siswa sekolah, pasien, atau orang tua yang membutuhkan perawatan dan pengawasan khusus. Hal ini membantu mengukur efektivitas pengobatan medis serta mendeteksi penanganan dini. VSN juga digunakan untuk telekonferensi dan pertemuan jarak jauh.
- Sistem *Telepresence*: Sistem *telepresence* berbasis VSN, pengguna dapat melihat setiap titik dari lokasi remote seolah-olah user benar-benar hadir secara fisik di lokasi tersebut.

### **2.2.3 Performansi Optimal *Visual Sensor Networks* dengan Mikroprosesor [4]**

Mikroprosesor maupun mikrokontroler termasuk kedalam penunjang kinerja untuk pengoptimalan VSN. Tingkat kinerja dari mikroprosesor dapat secara signifikan berdampak pada karakteristik disipasi daya node sensor. Hal ini dikarenakan mikroprosesor maupun mikrokontroler yang memiliki spesifikasi tinggi dan memiliki kinerja pengoperasian yang baik cenderung membutuhkan daya yang lebih

tinggi dibanding dengan tingkatan kualitas lainnya. Kinerja mikroprosesor maupun mikrokontroler bervariasi sesuai dengan pengaplikasiannya. Hal yang perlu diperhatikan untuk menciptakan kinerja VSN yang memenuhi tingkatan kinerjanya adalah pemilihan mikroprosesor yang tepat guna dan tidak berlebihan [5]. Hal ini untuk menunjang aspek performansi teknis maupun aspek non-teknis lainnya seperti biaya perancangan.

#### **2.2.4 Modus Operasi pada *Visual Sensor Networks***

Penghematan yang diupayakan dalam manajemen energi yang dibutuhkan dalam menjalankan prosesnya. Ada suatu fitur yang biasa digunakan dalam pengaplikasian WSN maupun VSN. Penghematan energi pada mikrokontroler maupun mikroprosesor biasanya memiliki beberapa modus operasi, diantaranya adalah modus aktif, kondisi *idle*, dan kondisi *sleep*. [8] Ketika sensor yang bertugas untuk mendeteksi suatu rangsangan atau pergerakan maka akan membutuhkan energi yang cukup banyak, namun ketika sensor dalam keadaan tidak mendeteksi rangsangan tetapi tetap melakukan *sensing* maka akan membutuhkan energi yang lebih banyak lagi.

Kondisi modus *idle* dan kondisi modus *sleep* sangat berpengaruh terhadap konsumsi energi dari suatu node dibandingkan suatu node harus selalu dalam kondisi aktif walaupun tidak ada suatu kejadian yang mengharuskan node itu untuk melakukan *sensing* terhadap objek. Semakin sering suatu node melakukan modus *sleep* ataupun modus *idle*, maka akan semakin kecil konsumsi energi yang dibutuhkan dan semakin besar penghematan yang dilakukan oleh node tersebut dalam sistem kerja WSN ataupun VSN. Selain itu, konsumsi daya pada tingkatan modus yang berbeda, waktu transisi, kekuatan transisi (*power transition*), dan total waktu yang dihabiskan oleh suatu mikroprosesor maupun mikrokontroler pada tiap node akan memiliki dampak yang signifikan terhadap total konsumsi energi yang dibutuhkan oleh masing-masing node sensor pada sistem kerja WSN maupun VSN.

#### **2.4 Pemrograman pada Video**

Pada ilmu matematika dan komputer, algoritma adalah prosedur langkah-demi-langkah untuk penghitungan. Algoritma digunakan untuk penghitungan, pemrosesan data, penalaran otomatis dan lain sebagainya. [5] Algoritma sangat penting untuk pengolahan data pada komputer. Banyak program komputer bergantung pada

algoritma yang memberikan rincian instruksi khusus untuk pelaksanaan kerjanya (dengan urutan tertentu). Maka sebuah algoritma bisa dianggap sebagai urutan operasi yang bisa disimulasikan oleh sebuah sistem lengkap.

#### **2.4.1 Pengertian Algoritma**

Algoritma adalah langkah-langkah yang disusun secara tertulis, berurutan (mempunyai tata urutan), untuk menyelesaikan suatu masalah. Pemrograman yang sederhana algoritma merupakan langkah pertama yang harus ditulis sebelum menuliskan program. Masalah yang dapat diselesaikan dengan pemrograman komputer adalah masalah-masalah yang berhubungan dengan perhitungan matematik.

#### **2.4.2 Teknik *Background Substraction***

Teknik algoritma *background subtraction*, juga dikenal sebagai *Foreground Detection*, adalah teknik di bidang pengolahan citra dan *computer vision* dimana latar depan gambar diambil untuk diproses lebih lanjut (*object recognition*) [10]. *Background subtraction* menjadi bagian yang sangat penting dari deteksi objek bergerak pada video. Masalah utamanya adalah ketepatan dalam proses menentukan objek bergerak. Teknik ini sangat tergantung terhadap seberapa sensitif atau tidaknya threshold yang ditentukan oleh program yang dibuat. Dengan nilai threshold merupakan selisih hasil dari determinan matriks yang diambil dari frame saat ini dengan frame yang dijadikan frame background. Tujuan utama dari proses deteksi objek bergerak dalam video adalah untuk mempartisi urutan gambar ke daerah yang berbeda kemudian dapat dilakukan pelabelan.

Teknik ini banyak digunakan dalam sistem pendeteksian terhadap pergerakan (*motion detection*) yang dimana mulai banyak diaplikasikan dalam beberapa teknologi terkait. Sistem kerja dari teknik *background subtraction* sangat tergantung pada besar *threshold* yang ditentukan dalam penggunaan teknik ini. Kesensitifan dari pergerakan benda dapat diukur dalam menentukan besarnya *threshold*.

Beberapa permasalahan dari deteksi objek yang dapat dikategorikan ke dalam tiga kategori menurut pendekatan utama masing-masing, diantaranya:

1. *Temporal differencing*
2. *Motion optical flow*
3. *Background subtraction*



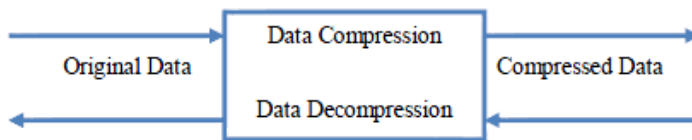
**Gambar 2.4 Contoh Event Algoritma Background Substraction**

Berdasarkan gambar diatas, gambar sebelah kiri merupakan *frame* yang dijadikan *background*. Pergerakan objek yang berupa seorang pria berjalan melewati background dan secara otomatis menjadi objek pergerakan yang terbaca dengan menggunakan teknik *background subtraction*. Terbacanya pergerakan pada *background* tersebut digambarkan sesuai dengan gambar disebelah kanan yaitu objek berupa seorang pria yang berjalan. Sifat dari teknik pendeteksian ini adalah menghitung perhitungan dari jumlah matriks yang ada pada *background frame* dengan matriks pada *foreground frame* dimana terdapat pergerakan objek yang melintas. Sehingga selanjutnya nilai dari selisih matriks tersebut akan bernilai suatu matriks selisih yang nantinya dilihat dari nilai *threshold* yang diatur. Apabila nilai selisih dari pergerakan matriks masuk kedalam nilai *threshold* yang ditentukan, maka dapat terdefinisikan sebagai suatu pergerakan.

### 2.4.3 Teknik Kompresi Video

Kompresi data merupakan cabang ilmu komputer yang bersumber dari teori informasi. Teori informasi menggunakan terminologi *entropy* sebagai pengukur berapa banyak informasi yang dapat diambil dari sebuah pesan. Ada empat pendekatan yang dapat digunakan pada kompresi suatu data, yaitu:

1. Pendekatan statistik. Contoh: *Huffman coding*
2. Pendekatan ruang. Contoh: *Run-Lenght encoding*
3. Pendekatan kuantisasi. Contoh: Kompresi kuantisasi (CS&Q)
4. Pendekatan Fraktal. Contoh: *Fractal image compression*



**Gambar 2.5 Proses Kompresi Data Secara Umum**

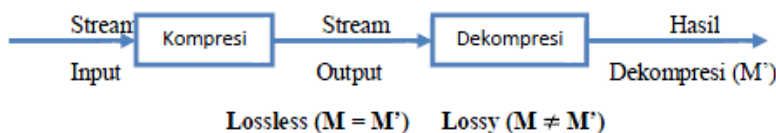
Teknik kompresi dibagi menjadi dua kategori besar, yaitu:

1. *Lossy Compression*

*Lossy compression* menyebabkan adanya perubahan data dibandingkan sebelum dilakukan proses kompresi. *Lossy compression* memberikan derajat kompresi lebih tinggi. Tipe ini cocok untuk kompresi file suara digital dan gambar digital. File suara dan gambar secara alamiah masih bisa digunakan walaupun tidak berada pada kondisi yang sama sebelum dilakukan kompresi.

2. *Lossless Compression*

*Lossless Compression* memiliki derajat kompresi yang lebih rendah tetapi dengan akurasi data yang terjaga antara sebelum dan sesudah proses kompresi. Kompresi ini cocok untuk basis data, dokumen atau spreadsheet. Pada Gambar 2.6 *lossless compression* ini tidak diijinkan ada bit yang hilang dari data proses kompresi,



**Gambar 2.6 Proses Kompresi Data Secara Lossless dan Lossy**

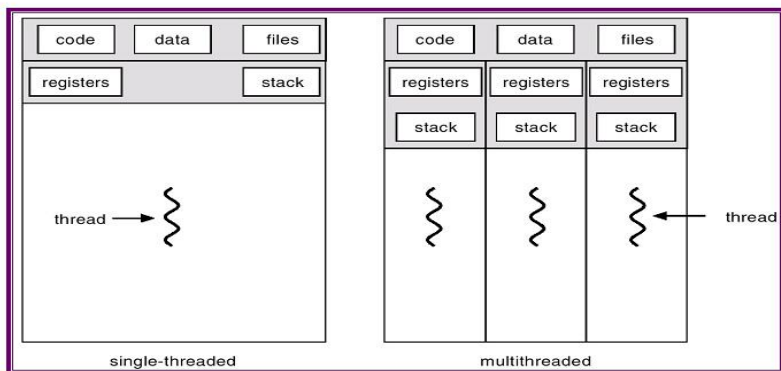
#### 2.4.4 Teknik *Multithread*

*Multithread* adalah suatu kemampuan yang memungkinkan beberapa kumpulan instruksi atau proses dapat dijalankan secara bersamaan dalam suatu program. Satu kumpulan instruksi yang akan dieksekusi secara independen dinamakan *thread*.

*Thread* adalah alur kontrol dari suatu proses atau sekumpulan instruksi yang dapat dilaksanakan/ dieksekusi secara teratur dengan

proses lainnya. Proses melakukan seriap langkah-langkah atau instruksi yang berurutan, setiap instruksi untuk mengeksekusi baris kode atau listing-listing program.

*Thread* sangat berguna untuk membuat proses yang interaktif; misalnya pada permainan atau game. Penggunaan sejumlah *thread*, membuat program tetap dapat menggerakkan sejumlah objek dan memberi kesempatan pemakai untuk melakukan tanggapan *thread*. Tanpa *thread*, *web browser* akan menghentikan segala tanggapan terhadap pemakai ketika perangkat lunak tersebut sedang mengambil isi dari suatu URL.



**Gambar 2.7 Ilustrasi Single dan Multi Thread**

Beberapa terminologi dari sistem kerja *multi thread* adalah:

- *Thread* Pengguna  
Pengaturannya dilakukan oleh pustaka *thread* pada tingkatan pengguna. Pustaka yang menyediakan fasilitas untuk pembuatan dan penjadwalan *thread*, *thread* pengguna cepat dibuat dan dikendalikan.
- *Thread* Kernel  
*Thread* yang didukung langsung oleh kernel. Pembuatan, penjadwalan, dan manajemen thread dilakukan oleh kernel pada kernel space. Dilakukan oleh sistem operasi, maka proses pembuatannya akan lebih lambat jika dibandingkan dengan *thread* pengguna.



Model-model *Multi Thread* antara lain:

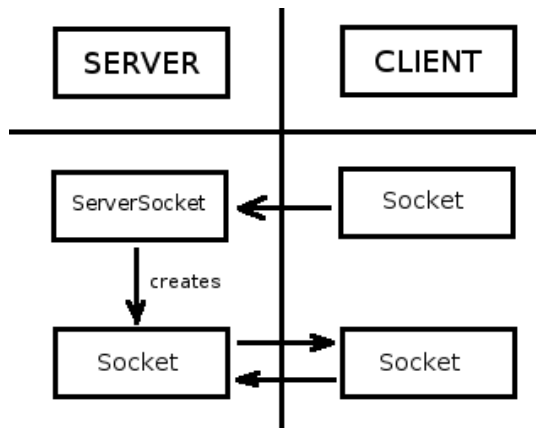
- Model Many-to-One.  
Model ini memetakan beberapa *thread* tingkatan pengguna ke sebuah *thread* tingkatan kernel. Pengaturan *thread* dilakukan dalam ruang pengguna sehingga efisien. Hanya satu *thread* pengguna yang dapat mengakses *thread* kernel pada satu saat. Jadi multi *thread* tidak dapat berjalan secara paralel pada multi prosesor.
- Model One-to-One.  
Model ini memetakan setiap *thread* tingkatan pengguna ke setiap *thread*, juga menyediakan lebih banyak *concurrency* dibandingkan model Many-to-One. Keuntungannya sama dengan *thread* kernel. Kelemahan model ini adalah setiap pembuatan *thread* pengguna membutuhkan tambahan *thread* kernel. Karena itu, jika mengimplementasikan sistem ini maka akan menurunkan kinerja dari sebuah aplikasi sehingga biasanya jumlah *thread* dibatasi dalam sistem.
- Model Many-to-Many.  
Model ini memultipleks banyak *thread* tingkatan pengguna ke *thread* kernel yang jumlahnya sedikit atau sama dengan pengguna. Model ini mengizinkan developer membuat *thead* sebanyak yang diinginkan tetapi *concurrency* tidak dapat diperoleh karena hanya satu *thread* yang dapat dijadwalkan oleh kernel pada suatu waktu. Keuntungan dari sistem ini adalah kernel *thread* yang bersangkutan dapat berjalan secara paralel pada multiprosesor.

#### 2.4.5 Teknik *Socket Communication*

Pada aplikasi *socket* yang sederhana diperlukan dua komponen penting penunjangnya. Pertama adalah aplikasi server yang akan menerima data, sedangkan yang kedua adalah aplikasi client yang akan mengirimkan data. Meskipun ada perbedaan peran dan fungsi pada keduanya, namun aplikasi server dan aplikasi client sama-sama mendefinisikan port yang sama sebagai jalur komunikasi.

Obyek *socket* pada sisi client dan server tetap ada perbedaan, Diantaranya adalah pada sisi aplikasi server, suatu *socket* server dibentuk dan melakukan operasi *listen* atau hanya menunggu. Operasi ini pada intinya menunggu permintaan koneksi dari sisi client yang akan mengirimkan data baik berupa data perintah ataupun data lainnya.

Informasi pada client alamat socket server dilewatkan sebagai argumen dan socket akan otomatis mencoba meminta koneksi ke socket server. Pada saat permintaan koneksi client telah diterima oleh server, maka socket server akan membuat suatu socket biasa. Socket ini akan berkomunikasi dengan socket pada sisi client. Setelah itu socket server dapat kembali melakukan listen untuk menunggu permintaan koneksi dari client lainnya. Setelah tercipta koneksi antara server dan client, selanjutnya keduanya akan dapat bertkar pesan dan sudah tersambung. Kedua aplikasi baik server maupun client dapat mengakhirir sambungan komunikasi dengan menutup socket.



**Gambar 2.8 Model Sistem Kerja Client-Server**

## 2.5 Mikroprosesor

Mikroprosesor (sering dituliskan  $\mu\text{P}$  atau  $\text{uP}$ ) adalah sebuah IC (*Integrated Circuit*) yang digunakan sebagai otak atau pengolah utama dalam sebuah sistem komputer. Sebuah mikroprosesor juga merupakan *Central Processing Unit* (CPU) elektronik komputer yang terbuat dari transistor mini dan sirkuit lainnya di atas sebuah sirkuit terintegrasi semikonduktor. Di dalam mikroprosesor terkandung rangkaian ALU (*Arithmetic-Logic Unit*), rangkaian CU (*Control Unit*) dan register-register. Rangkaian ALU berfungsi sebagai penyedia fungsi pengolahan. CU sebagai pengontrol fungsi prosesor. Register sebagai tempat penyimpanan sementara dalam mikroprosesor.

- ALU adalah alat yang melakukan pelaksanaan dasar seperti pelaksanaan aritmatika (penambahan, pengurangan, dan semacamnya), pelaksanaan logis (AND, OR, NOT), dan pelaksanaan perbandingan (misalnya, membandingkan isi sebanyak dua slot untuk kesetaraan). Pada unit inilah dilakukan “kerja” yang nyata.
- CU merupakan suatu alat pengontrolan yang berada dalam komputer yang memberitahu unit masukan mengenai jenis data, waktu pemasukan, dan tempat penyimpanan di dalam ALU mengenai operasi yang harus dilakukan, tempat data diperoleh, dan letak hasil ditempatkan. Perangkat-perangkat alat proses beserta perlengkapan
- Memory Unit (MU), merupakan bagian dari processor yang menyimpan alamat-alamat register data yang diolah oleh ALU dan CU.

### 2.5.1 Karakteristik Mikroprosesor

Karakteristik penting dari mikroprosesor adalah:

- Ukuran bus data internal (*internal data bus size*) merupakan jumlah saluran yang terdapat dalam mikroprosesor yang menyatakan jumlah bit yang dapat ditransfer antar komponen di dalam mikroprosesor
- Ukuran bus data eksternal (*eksternal data bus size*) merupakan jumlah saluran yang digunakan untuk transfer data antar komponen antara mikroprosesor dan komponen-komponen di luar mikroprosesor
- Ukuran alamat memori (*memory address size*) merupakan jumlah alamat memori yang dapat dialamati oleh mikroprosesor secara langsung
- Kecepatan clock (*clock speed*) rate atau kecepatan clock untuk menuntun kerja mikroprosesor
- Fitur-fitur spesial (*special features*) merupakan fitur khusus untuk mendukung aplikasi tertentu seperti fasilitas pemrosesan floating point, multimedia dan sebagainya

### 2.5.2 Komponen Dasar Sistem Mikroprosesor

Sistem mikroprosesor tersusun dari empat komponen dasar yaitu:

- Mikroprosesor
- Random Access Memory (RAM)
- Read Only Memory (ROM)
- Port Input/Output (PIO)

Pada proses pengoperasian, keempat komponen tersebut saling berkomunikasi/ mentransfer data. Media transfer datanya berupa sekelompok jalur-jalur penghubung yang disebut bus. Ada tiga jenis bus dalam sistem mikroprosesor, yaitu bus alamat, bus data, dan bus kontrol.

## 2.6 Raspberry Pi [3]

Raspberry pi sering juga disingkat dengan nama Raspi, adalah komputer papan tunggal (*Single Board Circuit/ SBC*) yang memiliki ukuran sebesar kartu kredit. Raspberry Pi bisa digunakan untuk berbagai keperluan, seperti spreadsheet, game, bahkan bisa digunakan sebagai media player karena kemampuannya dalam memutar video *high definition*. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation yang digawangi sejumlah developer dan ahli komputer dari Universitas Cambridge, Inggris. Ide dibalik komputer mungil ini diawali dari keinginan untuk mencetak generasi baru programmer.

Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum perbedaan yang mendasari antara kedua model Raspberry Pi keluaran tersebut adalah pada memori yang digunakan. Raspberry Pi model A menggunakan 256 RAM, sedangkan Raspberry Pi model B menggunakan 512 RAM. Perbedaan lainnya pada kedua model keluaran Raspberry Pi tersebut adalah Raspberry Pi model B terdapat kelebihan fasilitas lainnya berupa ethernet port yang tidak terdapat pada Raspberry Pi model A. Desain Raspberry Pi didasarkan seputar SoC (System-on-a-chip) Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, Video Core IV GPU, dan 256 Megabyte RAM pada Raspberry Pi model B. Penyimpanan data didesain tidak untuk menggunakan *hard disk* atau *solid-state drive*, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang.

Hardware Raspberry Pi tidak memiliki *real-time clock*, sehingga OS harus memanfaatkan timer jaringan server sebagai pengganti. Namun komputer yang mudah dikembangkan ini dapat ditambahkan dengan fungsi real-time (seperti DS1307) dan banyak lainnya, melalui saluran GPIO (General-purpose input/ output) via antarmuka I<sup>2</sup>C (Inter-Integrated Circuit).

Raspberry Pi bersifat *open source* (berbasis Linux), Raspberry Pi bisa dimodifikasi sesuai kebutuhan penggunaannya. Sistem operasi utama Raspberry Pi menggunakan Debian GNU/Linux dan bahasa

pemrograman Python. Salah satu pengembang OS untuk Raspberry Pi telah meluncurkan sistem operasi yang dinamai Raspbian. Raspbian diklaim mampu memaksimalkan perangkat Raspberry Pi, sistem operasi tersebut dibuat berbasis Debian yang merupakan salah satu distribusi Linux OS.

### **2.6.1 Spesifikasi Raspberry Pi B+**

Pada penelitian tugas akhir ini, tipe mikroprosesor yang digunakan adalah Raspberry Pi seri B+ sebanyak 2 buah. Spesifikasi dasar dari Raspberry Pi seri B+ adalah

- Prosesor ARM1176JZF-S 700 MHz
- RAM sebesar 512 MB
- SoC Broadcom BCM2835
- Ukuran 85,6 mm x 56,5 mm
- Terdapat 4 buah USB 2.0 HUB
- GPU VideoCore IV 250 MHz
- Micro SD on-board storage
- Penggunaan daya 600 mA (3,0 W)
- 10/100 Mbit/s Ethernet 8P8C



**Gambar 2.9 Raspberry Pi B+**

### 2.6.2 Desain Raspberry Pi Model B

Desain Raspberry Pi seputar SoC (System-on-a-chip) Broadcom BCM2835 telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU dan 512 Megabyte RAM (model B). Penyimpanan data didesain tidak untuk menggunakan *harddisk* atau solid-state drive, melainkan mengandalkan kartu SD (SD memory card) untuk booting dan penyimpanan jangka panjang. Raspberry Pi utamanya menjalankan sistem operasi berbasis kernel Linux. Sistem operasi utama Raspberry Pi menggunakan Desain GNU/Linux, mengemas Iceweasel, kaligrafi Suite dan bahasa pemrograman Python.

Pada penggunaannya Raspberry Pi telah menjelma menjadi suatu pernakat keras yang banyak digunakan untuk membuat atau menjalankan suatu program. Banyak pula peneliti yang mulai menggunakan Raspberry Pi sebagai komponen inti didalamnya.

## 2.7 Konsep Citra dan Video

Definisi citra adalah fungsi intensitas 2 dimensi  $f(x,y)$ , dimana  $x$  dan  $y$  adalah koordinat spasial dan  $f$  pada titik  $(x,y)$  merupakan tingkat kecerahan atau *brightness* suatu citra pada suatu titik. Suatu citra diperoleh dari pengakapan kekuatan sinar yang dipantulkan oleh objek. Citra sebagai output alat perekam, seperti kamera dapat bersifat analog maupun digital.

### 2.7.1 Konsep Citra

Pada dasarnya citra yang kita lihat merupakan sekumpulan cahaya yang ditangkap dengan indera mata kita dan dipersepsikan oleh otak menjadi obyek tertentu. Cahaya merupakan gelombang elektromagnetik. Suatu berkas cahaya dapat tersusun dari gelombang-gelombang panjang yang berbeda, dimana tiap panjang gelombang lainnya merepresentasikan jenis warna yang berbeda pula. Pada suatu media atau perangkat digital sering kita jumpai sehari-hari seperti kamera atau monitor setiap image atau citra digital direpresentasikan dengan karakteristik yang berbeda. Antara lain ukuran citra, resolusi, dan format nilainya. Pada umumnya citra digital berbentuk persegi panjang yang memiliki lebar dan tinggi tertentu. Ukuran ini biasanya dinyatakan dengan banyaknya titik atau *pixel* sehingga ukuran citra selalu bernilai bulat.

### **2.7.1.2 Penjejakan Obyek Bergerak**

Menjejak suatu obyek bergerak secara otomatis untuk mengetahui parameter dari obyek yang bergerak tersebut seperti kecepatan ataupun ukuran, merupakan aplikasi yang memakai teknik pengolahan citra. Pada dasarnya ada dua pendekatan yang dapat digunakan pada aplikasi ini, yaitu

- Penjejakan berdasarkan pengenalan obyek
- Penjejakan berdasarkan gerakan obyek

Sistem penjejakan obyek seperti yang diaplikasikan diberbagai pesawat militer untuk menjejak peluru kendali, dikembangkan dengan dasar pengembangan aplikasi prediksi gerakan. Pada obyek bergerak otomatis tersebut, dapat menjejak obyek yang bergerak pada area tertentu sendiri tanpa ada interverensi manusia. Pada penjejak obyek bergerak berdasarkan pengenalan obyek, bentuk atau pola obyek dikenali pada tiap-tiap frame dan penjejakan dilakukan dengan memberikan posisi dari obyek yang bergerak.

### **2.7.1.3 Kompresi Citra dan Video**

Perkembangan teknologi kompresi pada citra maupun video bereperanpada komunikasi multimedia dan berbagai aplikasi terkait lainnya. Transmisi citra ataupun video secara *realtime* tanpa kompresi pada komunikasi saat ini sulit dilakukan. Setiap aplikasi multimesia sangat membutuhkan kompresi citra atau video untuk menefisiensikan kebutuhan media penyimpanan dan juga bendwidht komunikasi.

### **2.7.3 Konsep Video**

Video merupakan gambar-gambar mati yang dibawa berurutan dalam suatu waktu dengan kecepatan tertentu. Gambar-gambar yang digabung tersebut dinamakan *frame* dan kecepatan pembacaan gambar disebut *frame rate*, dengan satuan fps (*frame per second*). Dimainkan dengan kecepatan yang tinggi maka tercipta ilustrasi gerak yang halus, semakin besar atau tinggi nilai satuan dari frame rate, maka akan semakin tinggi halus gambar yang dapat ditampilkan. Dalam video digital, pada dasarnya tersusun atas piksel-piksel yang tersusun menjadi array tiga dimensi. Dua dimensi menunjukkan gambar spasial (horizontal dan vertikal) dan satu dimensi lainnya merepresentasikan waktu atau jumlah gambar. Pada video digital, frame merupakan kumpulan piksel-piksel yang menunjukkn titik tertentu dalam satu waktu.

### 2.7.3.1 Video Analog dan Digital

Video analog tersusun dari gelombang bersambung yang bervariasi, dengan kata lain nilai sinyal akan memiliki angka yang beragam tetapi pada batas maksimum dan minimum yang diijinkan. Digital video ditransmisikan hanya berupa titik presisi yang dipilih pada interval dalam kurva. Tipe sinyal digital yang dapat dipakai oleh komputer kita adalah tipe binary. Data biner diwakilkan dengan angka 1 dan 0. Angka 1 mewakili nilai maksimum dan angka 0 mewakili nilai minimum.

Video digital memiliki beberapa kelebihan dibandingkan dengan analog video, diantaranya adalah

- Ketepatan yang tinggi dalam hal transmisi (*high fidelity*) dibandingkan dengan sinyal analog. Pada sinyal analog, saat penerima akhir transmisi akan sulit membedakan sinyal asli dan *noise* yang mungkin diperkenalkan selama transmisi. Transmisi yang diulang-ulang maka akumulasi *noise* tidak dapat dihindari. Lain halnya dengan sinyal digital yang dapat membedakan antara sinyal asli dan *noise*. Sinyal digital juga dapat ditransmisikan berulang-ulang sebanyak yang kita inginkan tanpa mempengaruhi kualitasnya.
- Mudah untuk diproses, perbedaan antara video analog dan digital dapat dianalogikan dengan membandingkan antara mesin ketik dengan pengolahan kata di komputer. Mengetik di komputer memudahkan dan mempercepat kita dalam menghapus dan menambah kata. Selain itu, dalam video digital kita dapat memisahkan gambar dan suara tanpa saling mempengaruhi.
- Penyimpanan data dalam video digital dapat dilakukan berulang-ulang tanpa ada informasi-informasi penting yang terkandung dalam media penyimpanan dapat hilang.
- Mudah dienkripsi dan lebih baik dalam menghadapi *noise* kanal.

Dunia video kini telah mengalami perubahan dari analog ke digital. Perubahan ini terjadi pada setiap tingkatan industri. Pada konsumen rumahan dan perkantoran kita dapat menikmati kualitas video digital yang prima lewat hadirnya teknologi VCD dan DVD, sedangkan dunia broadcasting juga sudah mulai beranjak kepada teknologi DTV (*Digital Television*).

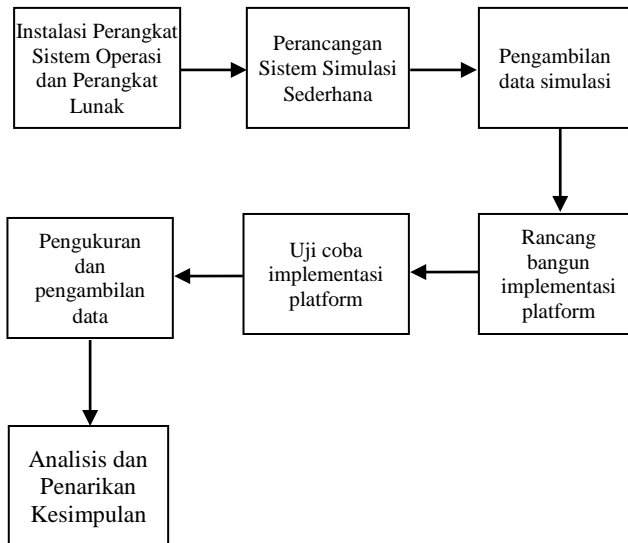


### BAB 3

## PERANCANGAN DAN IMPLEMENTASI

Tujuan dari tugas akhir ini adalah untuk merancang sistem dan algoritma pada VSN yang dapat menghemat energi dan meminimalisi penyimpanan data video output yang berlebih pada suatu kejadian real time. Pengujian algoritma akan dilakukan sebanyak 3 kali, dalam kondisi berupa pengambilan gambar dengan algoritma pada kondisi ruangan *indoor* dan terdapat objek pemodelan. Hasil yang diperoleh akan dibandingkan antara kedua hasil video output. Kemudian dibandingkan yang lebih efektif dan lebih efisien dalam penghematan energi beserta kondisi dan syarat terjadinya event.

Di dalam bab ini juga dibahas tentang hal-hal yang berkaitan dengan perancangan algoritma yang digunakan dalam tugas akhir ini serta langkah-langkah yang dilakukan untuk memperoleh hasil yang ditentukan. Alur penelitian Tugas akhir ini dapat dilihat pada Gambar 3.1.



**Gambar 3.1 Diagram Alur Penelitian**

### 3.1 Perancangan Uji Unjuk Kerja Sistem

Hal yang diperlukan untuk mendapatkan efisiensi energi dan memori yang dipakai adalah suatu algoritma yang dapat memperkecil dan mengoptimalkan konsumsi energi dan konsumsi data yang tersimpan. Pada bab ini dijelaskan tentang simulasi sistem kerja, perancangan algoritma, desain algoritma, dan implementasi VSN dengan menggunakan mikroprosesor Raspberry Pi seri B+.

Langkah-langkah yang digunakan untuk perancangan simulasi sederhana yaitu:

1. Simulasi sederhana sistem menggunakan program Matlab R2011b
2. Instalasi perangkat lunak pendukung (Matlab R2011b)
3. Melakukan pemrograman dengan mendeteksi pergerakan objek dan menentukan koordinat dari objek
4. Masukan video input, dan akan terdeteksi pergerakan juga koordinat objek.

Metode pengujian simulasi sederhana menggunakan Matlab R2011b yaitu:

1. Simulasi sistem kerja sederhana deteksi pergerakan
2. Pendeteksian pergerakan dilakukan dengan menggunakan fungsi algoritma *background subtraction*
3. Setelah itu masukan video inputan dan jalankan fungsi pada m.file pada Matlab R2011b.
4. Pengukuran pada simulasi sederhana hanya difokuskan pada pendeteksian gerakan. Karena pengukuran lainnya akan dijalankan pada percobaan implementasi
5. Setelah terdeteksi objek yang berjalan, maka selanjutnya adalah menentukan koordinat pada pixel berapa objek tersebut dapat terbaca oleh program.
6. Setelah itu kita dapat melaksanakan pengolahan data dan Penarikan kesimpulan

Langkah-langkah perancangan implementasi pada kamera yaitu:

1. Implementasi sistem kerja VSN pada tugas akhir ini dilakukan dengan menggunakan bantuan mikroprosesor Raspberry Pi
2. Diperlukan instalasi Sistem Operasi (OS) dari Raspberry Pi, pada tugas akhir ini digunakan sistem operasi Raspberry Pi berupa Raspbian yang dijalankan pada Linux.

3. Setelah instalasi sistem operasi, maka selanjutnya adalah instalasi perangkat lunak pendukung lainnya seperti OpenCV dan bahasa pemrograman C++
4. Setelah seluruh perangkat lunak pendukung lainnya terinstalasi maka dapat dilakukan pemrograman untuk algoritma VSN

Metode pengujian implementasi platform yaitu:

1. Setelah melakukan instalasi pada sistem operasi dari Raspberry Pi, dan juga telah dilakukan instalasi terhadap perangkat lunak pendukung lainnya, maka dapat dilakukan pengkodean untuk membuat program berupa algoritma sistem kerja VSN.
2. Pemanfaatan 2 kamera webcam dan 2 mikroprosesor Raspberry Pi, membuat sistem untuk mengukur daya yang dipakai oleh kedua mikroprosesor Raspberry Pi dan kamera webcam, juga mengukur pemakaian CPU Usage pada Raspberry Pi.
3. Selanjutnya dibandingkan antara pengukuran video yang dihasilkan dari sistem VSN yang menggunakan algoritma hemat energi dengan tanpa menggunakan algoritma.
4. Pengambilan kesimpulan dari perbandingan antar keduanya yang lebih efisien untuk diimplementasikan.

### **3.2 Kebutuhan Pendukung Infrastruktur**

Kebutuhan pendukung infrastruktur pada perancangan dan implementasi sistem *Visual Sensor Networks* ini terbagi menjadi dua macam, yaitu *software* dan *hardware*, dimana keduanya saling mendukung satu sama lain.

#### **3.2.1 Perangkat keras (Hardware)**

Perangkat keras yang digunakan dalam implementasi sistem sinkronisasi di tugas akhir ini yaitu:

1. Raspberry Pi seri B+, Raspberry Pi seri B+ yang digunakan pada sistem kerja ini membutuhkan minimal 2 buah. Salah satu Raspberry Pi seri B+ akan bertindak sebagai server yang menerima pengiriman paket data berupa perintah untuk berkomunikasi secara serial dengan command tertentu serta dapat mengirimkan paket ping. Raspberry Pi seri B+ lainnya bertindak sebagai client yang bertugas sebagai pengirim command untuk sinkronisasi komunikasi dalam menjalankan program. Dalam tugas akhir ini Raspberry Pi seri B+ direpresentasikan sebagai mikroprosesor pengendali dan yang

menjalankan program algoritma. Spesifikasi dari Raspberry Pi seri B+ adalah:

- Prosesor ARM1176JZF-S 700 MHz
  - RAM sebesar 512 MB
  - SoC Broadcom BCM2835
  - Ukuran 85,6 mm x 56,5 mm
  - Terdapat 4 buah USB 2.0 HUB
  - GPU VideoCore IV 250 MHz
  - Micro SD on-board storage
  - Penggunaan daya 600 mA (3,0 W)
  - 10/100 Mbit/s Ethernet 8P8C
2. *Webcam* m-tech berperan sebagai sensor dan kamera perekam. Fungsinya sangat penting bagi sistem kerja VSN. Spesifikasi dari webcam m-tech yang digunakan adalah:
- Material : Metal body, Plastic leg, berat : 125 gram
  - Built up 1.3 Megapixel Resolution
  - No Driver Required (Windows XP SP2/Vista/7)
  - USB 2.0 Interface
  - Panjang kabel USB : 140-145 cm
  - Lensa : F/#2.0 F:4.8mm.OC
  - Manual Fokus
  - Berat : 125 g



**Gambar 3.2 Webcam M-Tech 1.3 Megapixel**

3. Micro SD Card. Dikarenakan keterbatasan infrastruktur yang ada pada mikroprosesor Raspberry Pi, maka diperlukan infrastruktur

tambahan untuk Raspberry Pi yaitu memory card *external* untuk penyimpanan dan proses *booting* dalam membantu proses perancangan implementasi sistem. Micro SD berperan sebagai memori penunjang sistem.

4. Kabel Power Supply. Spesifikasi kabel power supply sama seperti power supply yang digunakan pada smartphone pada umumnya. Power supply berperan sebagai pembangkit daya dari Raspberry Pi seri B+, dengan arus input 1,2A.
5. Monitor Acer 17" berperan sebagai output tampilan pada mikroprosesor Raspberry Pi B+ yang digunakan. Raspberry Pi tidak memiliki outputan untuk menampilkan tampilannya sendiri, sehingga memerlukan tambahan.
6. Keyboard M-Tech digunakan untuk memberikan input dalam pemrograman menggunakan bahasa pemrograman yang digunakan. Raspberry Pi membutuhkan tambahan inputan dalam menunjang fungsinya sebagai mikroprosesor.
7. Mouse Logitech merupakan infrastruktur tambahan dalam penunjang Raspberry Pi, karena Raspberry Pi tidak memiliki mouse pad atau infrastruktur pointer lainnya.
8. Kabel Connector HDMI to VGA digunakan untuk mengkonversi port yang ada pada Raspberry Pi kepada port yang tersedia pada monitor.
9. Kabel RJ45. Kabel Ethernet digunakan dalam proses SSH antara kedua Raspberry Pi untuk melakukan komunikasi dan sinkronisasi dalam pengambilan data.

### **3.2.2 Perangkat Lunak (*Software*)**

Perangkat lunak digunakan untuk membuat dan menjalankan program data dan mengakses data. Sama halnya dengan perangkat keras, perangkat lunak juga mempunyai peranan yang sangat penting dalam implementasi sistem kerja pada tugas akhir ini. Spesifikasi perangkat lunak yang digunakan adalah:

#### **3.2.2.1 *Matlab R2011b***

Matlab (*matrix laboratory*) adalah sebuah lingkungan komputasi numerikal dan bahasa pemrograman komputer generasi keempat. Matlab dapat memungkinkan manipulasi matriks, peletakan fungsi dan data, implementasi algoritma, pembuatan antar muka pengguna, dan peng-antarmuka-an dengan program dalam bahasa lainnya. Meskipun hanya bernuansa numerik, sebuah kotak kakas

(*toolbox*) yang menggunakan mesin simbolik MuPAD, memungkinkan akses terhadap kemampuan aljabar komputer. Sebuah paket tambahan, Simulink. Menambahkan simulasi grafis multiranah dan Desain Berdasar-Model untuk sistem terletak dan dinamik.

#### **3.2.2.2 OpenCV**

OpenCV (*Open Source Computer Vision Library*) adalah sebuah pustaka perangkat lunak yang ditujukan untuk pengolahan citra dinamis secara real-time, yang dibuat oleh intel, dan sekarang didukung oleh Willow Garage dan Itseez. Program ini bebas dan berada dalam naungan sumber terbuka dari lisensi BSD. Pustaka ini merupakan pustaka litar platform. Program ini didedikasikan sebagian besar untuk pengolahan citra secara real-time. Jika pustaka ini menemukan pustaka Integrated Performance Primitives dari intel dalam sistem komputer, maka program ini akan mempercepat proses kerja program secara otomatis.

#### **3.2.2.3 Putty**

Putty adalah sebuah program open source yang dapat digunakan untuk melakukan protokol jaringan SSH, Telnet dan Rlogin. Protokol ini dapat digunakan untuk menjalankan sesi remote pada sebuah komputer melalui sebuah jaringan, baik itu LAN, maupun internet. Putty digunakan untuk memprogram menggunakan SSH untuk uji coba menghubungkan 2 mikroprosesor Raspberry Pi menggunakan algoritma Socket dan Multithread.

#### **3.2.2.4 C++**

Bahasa C++ adalah pengembangan dari bahasa pemrograman C yang diciptakan oleh Dennis MacAlistair Ritchie. Setelah itu dikembangkan oleh Bjarne Stroustrup menjadi C++. Sementara itu kedudukan bahasa C/C++ ini adalah bahasa aras menengah yaitu diantara bahasa aras rendah seperti Assembly yaitu bahasa mesin, dan bahasa aras tinggi seperti java, pascal dll, yang lebih dekat dengan bahasa manusia. Kelebihan bahasa C++ yaitu:

- Penggunaan bahasa pemrogramannya mendekati pada bahasa manusia
- Memiliki kompatibilitas yang tinggi antar platform
- Untuk menulis pemrograman C++ dapat menggunakan sembarang aplikasi text editor seperti notepad, notepad++, wordpad, Borland C++, dan lainnya.

Bahasa pemrograman C++ ini digunakan sebagai bahasa pemrograman utama untuk membuat sistem dan merancang algoritma untuk VSN.

### **3.2.2.5 LX Terminal**

LXTerminal adalah emulator berbasis VTE standar terminal untuk LXDE (Lightweight X Desktop Environment) tanpa ketergantungan yang tidak perlu. Terminal telah mendapat fitur bagus seperti Fitur LXTerminal

- Banyak tab dukungan
- Mendukung perintah umum seperti `cd`, `dir`, `mkdir`, `mkdir`
- Fitur untuk menyembunyikan menu bar untuk ruang
- Mengubah skema warna

LXTerminal ini digunakan sebagai pemanggil algoritma yang telah dibuat di pemrograman C++.

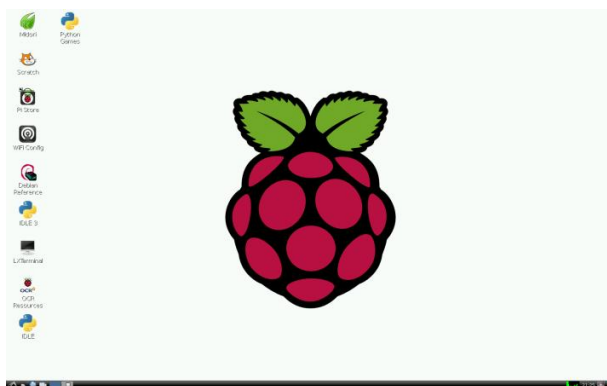
## **3.3 Instalasi Perangkat Lunak (Software)**

Berbagai komponen-komponen yang ada dalam tugas akhir ini, diperlukan software atau perangkat lunak yang spesifik untuk menunjang pembuatan sistem. Sebelum melakukan perancangan sistem dan perancangan algoritma terlebih dahulu harus dilakukan instalasi software.

### **3.3.1 Instalasi Operational System Raspberry Pi**

Sebelum mengaktifkan dan menjalankan Raspberry Pi perlu dilakukan instalasi sistem operasi. Berikut merupakan langkah dalam melakukan instalasi terhadap sistem operasi Raspberry Pi. Pengunduhan sistem operasi Raspbian salah satu caranya adalah dengan mengunduhnya di [raspberrypi.org/downloads](https://www.raspberrypi.org/downloads). Penginstallan sistem operasinya, dibutuhkan pula Micro SD card dengan kapasitas minimal 4GB dengan spesifikasi kelas 10. Micro SD diperlukan untuk menyimpan data sebagai pengganti Harddisk pada komputer biasa. Penggunaan media penyimpanan seperti Micro SD bisa berbeda-beda jenisnya tergantung seri Raspberry Pi yang kita gunakan, untuk seri B+ digunakan Micro SD card. Proses instalasi tidak terlalu sulit, banyak panduan untuk menginstal sistem operasi dari Raspberry Pi ini.

Jika Raspberry Pi telah sukses diinstal, maka langkah selanjutnya kita dapat menjalankan Raspberry Pi seperti kita menjalankan komputer pada umumnya. Karena Raspberry Pi merupakan *mini computer* maka ada kekurangan pada Raspberry Pi. Penjalanan fungsinya Raspberry Pi membutuhkan perangkat keras lainnya seperti monitor dengan adapter HDMI (High-Definition Multimedia Interface), keyboard USB dan mouse USB ke port USB milik Raspberry Pi, dan keperluan hardware lainnya.



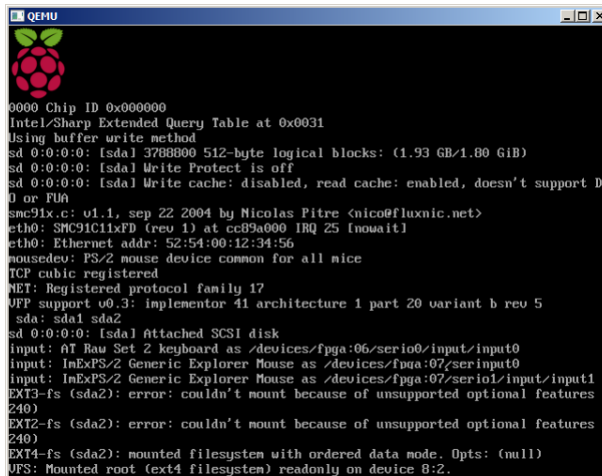
**Gambar 3.3 Tampilan Desktop Raspberry Pi**

Menyalakan Raspberry Pi dibutuhkan adapter untuk daya dengan tegangan 5 V dan arus kisaran 1,0-1,5 A. Rentang daya yang diberikan harus tetap pada *range* yang telah ditentukan tersebut. Apabila daya sudah berhasil menghidupkan Raspberry Pi, maka secara otomatis Raspberry Pi akan melakukan *booting*. Setelah Raspberry berhasil melakukan *booting* maka selanjutnya Raspberry Pi akan masuk kedalam tampilan menu seperti pada gambar diatas.

Hal yang perlu diperhatikan adalah untuk pertama kali menggunakan Raspberry Pi, ketika selesai *booting* user akan diminta untuk memasukan User ID dan Password. Diawal setting Use ID adalah “pi” dan setting password adalah “raspberry”.

Namun untuk *setting up* dari password dapat diganti untuk meningkatkan keamanan dari mikroprosesor Raspberry Pi itu sendiri. Raspberry Pi. Selain itu, dalam mikroprosesor Raspberry Pi dapat pula digunakan user lain selain Pi. Seperti user *Root* dan lainnya dimana tiap user disini punya fungsi yang sama seperti fungsi yang lainnya.





```
QEMU
0000 Chip ID 0x000000
Intel/Sharp Extended Query Table at 0x0031
Using buffer write method
sd 0:0:0:0: [sdal 3788800 512-byte logical blocks: (1.93 GB/1.80 GiB)
sd 0:0:0:0: [sdal Write Protect is off
sd 0:0:0:0: [sdal Write cache: disabled, read cache: enabled, doesn't support DP
U or FUA
smc91x.c: v1.1, sep 22 2004 by Nicolas Pitre <nico@fluxnic.net>
eth0: SMC91C11xFD (rev 1) at cc89a000 IRQ 25 [nowait]
eth0: Ethernet addr: 52:54:00:12:34:56
mousedev: PS/2 mouse device common for all mice
TCP cubic registered
NET: Registered protocol family 17
UFP support v0.3: implementor 41 architecture 1 part 20 variant b rev 5
sda: sda1 sda2
sd 0:0:0:0: [sdal Attached SCSI disk
input: AT Raw Set 2 keyboard as /devices/fpga:06/serio0/input/input0
input: ImEXPS/2 Generic Explorer Mouse as /devices/fpga:07/serio0/input/input0
input: ImEXPS/2 Generic Explorer Mouse as /devices/fpga:07/serio1/input/input1
EXT3-fs (sda2): error: couldn't mount because of unsupported optional features (
240)
EXT2-fs (sda2): error: couldn't mount because of unsupported optional features (
240)
EXT4-fs (sda2): mounted filesystem with ordered data mode. Opts: (null)
VFS: Mounted root (ext4 filesystem) readonly on device 8:2.
```

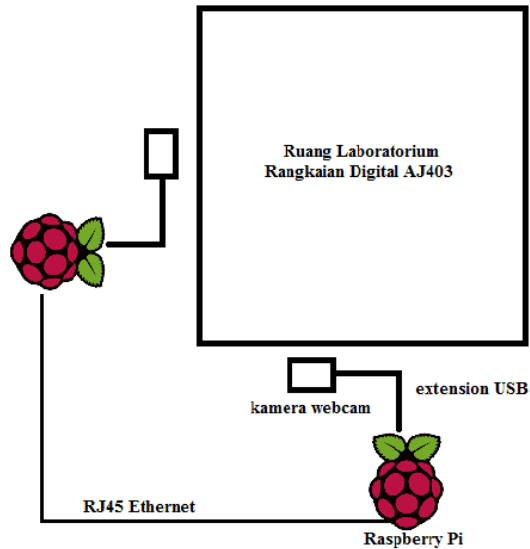
**Gambar 3.4 Tampilan Booting Raspberry Pi**

### 3.4 Desain Sistem Kerja

Sistem ini bertujuan untuk membuat sistem kerja dan algoritma dari VSN yang hemat energi dan mengkonsumsi kapasitas memori yang sedikit. Desain sistem yang dirancang menggunakan mikroprosesor Raspberry Pi seri B+ sebagai pengatur dan pusat koordinator dalam memproses data serta menjalankan algoritma. [6] Komponen dari sistem kerja yang terdiri dari mikroprosesor Raspberry Pi, kamera *webcam*, dan perangkat pendukung mikroprosesor lainnya akan saling bekerja sama untuk mengambil gambar dari objek yang tertangkap dan memprosesnya serta menyimpan hasil rekaman citra yang ditangkap. Untuk menunjang lokasi penempatan kamera yang berada jauh dari *port USB* Raspberry Pi, maka dapat digunakan kabel tambahan perpanjangan (*Extention USB Cable*) yang dapat menghubungkan antara kamera dan Raspberry Pi.

#### 3.4.1 Skema dan Sistem Kerja *Visual Sensor Networks*

VSN yang bekerja mempunyai skema seperti diatas. Kamera diletakan pada dua sisi dinding pada ruangan LaboratoriumAJ403. Menghadap pada lokasi yang mempunyai beberapa daerah. Karena kamera menghadap daerah yang sama, maka ada daerah yang termasuk kedalam daerah overlap. Daerah overlap adalah daerah dimana lokasi masuk kedalam daerah *coverage* kedua kamera tersebut.

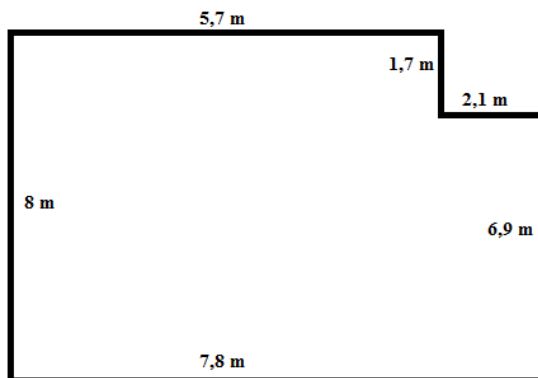


**Gambar 3.5 Skema Kerja *Visual Sensor Networks***

Ketika objek melintas pada daerah yang termasuk daerah jangkauan atau daerah coverage salah satu atau kedua kamera tersebut, maka secara otomatis kamera akan melakukan suatu perintah berupa perekaman pergerakan dan objek tersebut. [7] Karena kamera bekerja dengan mengandalkan mikroprosesor dan mikroprosesor Raspberry Pi yang digunakan 2 buah, maka proses perekaman harus bersamaan dan real time. Memanfaatkan fungsi dari SSH antara kedua mikroprosesor dan algoritma socket dan multi *thread* maka proses pengambilan gambar dapat dilakukan secara bersamaan pada waktu yang sama. Sehingga antara kamera client dan kamera server tidak terdapat perbedaan waktu dalam perekaman dan hasil dari perekaman tetap pada waktu yang sama antara tiap-tiap node.

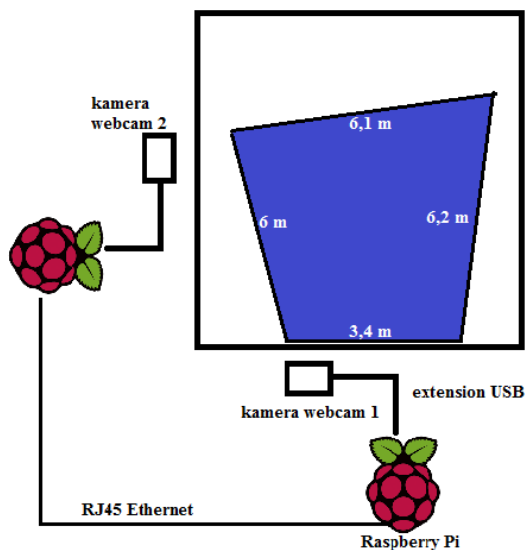
### 3.4.2 Lokasi Pengimplementasian *Visual Sensor Networks*

Lokasi yang digunakan dalam pengambilan data adalah Ruang Laboratorium Rangkaian Digital AJ-403 Jurusan Teknik Elektro ITS Surabaya. Lokasi ini dijadikan lokasi implementasi VSN. Ruangan Laboratorium Rangkaian Digital AJ-403 ini memiliki keliling 32,2 m dan luas 63,51 m<sup>2</sup>.



**Gambar 3.6 Denah Ruang Laboratorium AJ403**

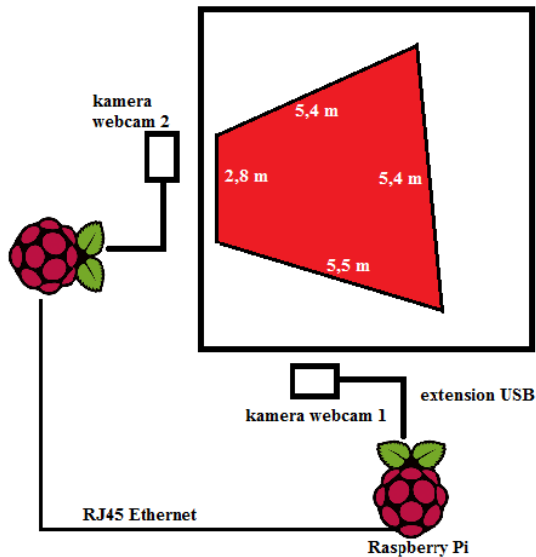
Saat proses pengambilan gambar dari suatu sudut diperlukan area daerah jangkauan dari masing-masing kamera tersebut. Hal ini untuk mengukur daerah yang termasuk kedalam daerah overlap.



**Gambar 3.7 Daerah Coverage Kamera 1**

Gambar 3.7 merupakan gambar daerah cakupan dari kamera 1. Kamera satu berfungsi sebagai kamera server, tugasnya menerima pengiriman data untuk melakukan perintah *recording*. Setiap kamera mempunyai sudut pandang menyerupai bentuk trapesium. <sup>2</sup>

Sementara daerah jangkauan dari kamera 2 seperti pada gambar dibawah ini. Kamera client merupakan node yang bertugas untuk mengirimkan perintah untuk melakuka proses perekaman yang terintegralisasikan.



**Gambar 3.8 Daerah Coverage Kamera 2**

### 3.5 Algoritma Implementasi Visual Sensor Networks

Setelah melakukan instalasi perangkat lunak pendukung desain sistem kerja, maka selanjutnya kita dapat memulai untuk proses pemrograman untuk mendapatkan sistem kerja algoritma. Pada subbab ini akan dibahas mengenai fungsi masing-masing dari algoritma pemrograman. Tiap program mempunyai fungsi yang berbeda-beda sesuai dengan perintah yang diberikan. Perancangan program ini, fungsi yang diberikan harus cocok antar satu perintah kepada perintah yang lain, agar tidak terjadi bentrok perintah di dalamnya. Suatu teknik

deteksi obyek pada video digital, terdapat beberapa pendekatan yang dapat dijalankan, salah satunya yaitu memakai teknik *background subtraction* sebagai dasar dalam pengolahan citra. Teknik ini juga terdapat bermacam-macam sesuai dengan kemampuannya.

### 3.5.1 Simulasi Kerja Sederhana

Pada tugas akhir ini, dibuat simulasi sederhana dengan menggunakan software Matlab edisi R2011b. Pada simulasi sederhana didapatkan cara kerja untuk mendeteksi pergerakan dari suatu objek bergerak dengan menggunakan webcam. Setelah mengetahui pergerakan benda tersebut, dilakukan pendeteksian koordinat objek pada lokasi medan nyata. Berikut ini adalah cuplikan program dari simulasi sederhana

```
%Read Background Image
Background=imread('background.jpg');
%Read Current Frame
CurrentFrame=imread('original.jpg');
%Display Background and Foreground
% subplot(2,2,1);imshow(Background);title('BackGround');
% subplot(2,2,2);imshow(CurrentFrame);title('Current Frame');

Background_gray=rgb2gray(Background);
CurrentFrame_gray=rgb2gray(CurrentFrame);

[rows columns]=size(Background_gray);
```

Setelah didapatkan deteksi pergerakan pada frame yang diinginkan, maka selanjutnya adalah menentukan lokasi dan koordinat daripada objek tersebut. Penentuannya dari cuplikan gambar dan menggunakan algoritma sebagai berikut,

```
[m,n]=size(BinaryImage_dilate);
a=[];
b=[];
```

```

for i=1:m;
    for j=1:n;
        if BinaryImage_dilate(i,j)==1;
            a=[a i];
            b=[b j];
        end
    end
end
loc=size(a,2);
pointy=a(loc);
pointx=b(loc);

```

### 3.5.2 Teknik *Background Subtraction*

Berdasarkan penjelasan diatas maka perintah yang digunakan untuk melakukan pendeteksian objek yang bergerak adalah:

```

///background subtraction
absdiff(back, gray_frame, diff);

erode(diff, diff, kernel_ero);

int nilai = countNonZero(diff);

if (nilai > 1000)
{

imshow("frame", frame);

```

Seperti pada fungsi utama dari teknik background subtraction pada dasarnya, disini teknik background subtraction difungsikan sebagai pendeteksian gerakan yang ditangkap oleh kamera. Secara

umum algoritma yang digunakan untuk mendeteksi obyek bergerak adalah dengan berbagai macam teknik *background subtraction*. Teknik ini bertujuan untuk menentukan kemungkinan adanya objek yang bergerak, yaitu dengan mendapatkan frame-frame dari suatu kamera diam yang berisi obyek *foreground* atau dengan kata lain memisahkan background dan foreground dari suatu gambar.

Pendekatan sederhana yang dapat dijalankan untuk mencapai tujuan tersebut yaitu dengan membedakan atau mendapatkan selisih antara frame tertentu dengan frame yang berisi gambar statis, atau dapat dituliskan

$$| \text{frame } i - \text{background} | > Th$$

Masalah pertama yang dihadapi dalam teknik ini yaitu bagaimana mendapatkan suatu gambar statis yang merepresentasikan background secara otomatis.

Penanganan masalah tersebut hal yang harus diperhatikan yaitu background yang didapat tidak tetap tetapi harus dapat beradaptasi dengan adanya:

1. Perubahan iluminasi, baik secara gradual maupun tiba-tiba (seperti perubahan cahaya akibat pergerakan awan)
2. Perubahan gerakan, baik karena kamera yang bergoyang maupun adanya pergerakan obyek pada background (seperti pergerakan batang pepohonan, aliran air pada sungai atau laut dan lainnya)
3. Perubahan bentuk background, seperti adanya mobil yang parkir dan sebaliknya.

Metode dasar yang bisa digunakan adalah menggunakan selisih antar frame, atau dapat dituliskan

$$| \text{frame } i - \text{frame } i-1 | > Th$$

Dalam metode selisih frame ini estimasi *background* adalah berasal dari frame sebelumnya. Metode ini hanya akan bekerja dengan baik pada kondisi kecepatan obyek dan frame rate tertentu. Untuk mendapatkan hasil yang baik, nilai *Th* (*Threshold*) harus diubah-ubah atau dengan kata lain nilai *Th* sangat sensitif.

### 3.5.3 Teknik Kompresi

Meneruskan dari subsub bab 3.5.2, algoritma yang digunakan bertujuan untuk mengompresi resolusi yang pada gambar yang ditangkap oleh kamera dengan gambar yang disimpan pada mikroprosesor. Hal ini memungkinkan untuk dilakukan karena apabila kamera mengawasi daerah yang terbilang sangat ramai dan banyak pergerakan didalamnya, yang secara otomatis akan terus menerus ditangkap oleh kamera, maka akan sulit dilakukan penghematan.

Mengapa dilakukan kompresi resolusi, karena kita ketahui bahwa video digital memiliki ukuran yang dinamakan resolusi (dinyatakan dalam panjang x lebar), 320 x 240 piksel atau 640 x 480 piksel dan sebagainya. Semakin besar resolusi video maka ukuran bidang gambar yang dapat ditampilkan semakin besar namun ukuran file video tersebut juga akan semakin besar.

Selain resolusi, faktor bitrate atau bandwidth juga memegang peranan dimana nilai ini menentukan seberapa banyak data yang dibutuhkan untuk memainkan video tersebut per detik. Semakin besar bitratnya maka semakin tinggi kualitas video digital tersebut. Resolusi yang besar ditambah bit rate yang tinggi membuat sebuah file video “mentah” memiliki ukuran sangat besar. Ada faktor yang tidak kalah penting dalam urusan video digital yaitu teknik kompresi video. Teknik ini merupakan proses matematis rumit yang sebisa mungkin sama baiknya seperti video “mentah” yang tidak terkompresi. Idealnya kompresi yang baik mampu memberi hasil yang sebaik mungkin dengan ukuran yang sekecil mungkin.

**Tabel 3.1 Tabel Pengelompokkan Resolusi dan Frame Rate**

| Use Scenario                   | Resolution & Frame Rate | Example Data Rates |
|--------------------------------|-------------------------|--------------------|
| Mobile Content                 | 176x144, 10-15 fps      | 50-60 Kbps         |
| Internet / Standart Definition | 640x480, 24 fps         | 1-2 Mbps           |
| High Definition                | 1280x720, 24p           | 5-6 Mbps           |
| Full High Definition           | 1920x1080, 24p          | 7-8 Mbps           |



Berikut ini adalah algoritma dari kompresi:

```
///kompresi resolusi

        resize(frame,frame,Size(frame_width/2,frame_height/2),0,0,IN
TER_CUBIC);

                video1.write(frame);

        }

        imshow("frame", frame);

                char c = cvWaitKey(1);
                if (c == 27) break;

        }

        //return 0;
}
```

Algoritma kompresi disini berfungsi sebagai penghematan dalam memori. Cara kerja algoritma ini adalah dengan mereduksi resolusi menjadi setengahnya dari gambar yang ditangkap oleh kamera. Apabila resolusi yang berhasil ditangkap oleh kamera adalah 640 x 480 piksel, namun karena kamera diberi algoritma berupa kompresi resolusi antara gambar yang ditangkap dengan video yang disimpan maka pada Raspberry Pi video yang disimpan dari proses pengambilan gambar akan memiliki resolusi setengah dari yang ditangkap oleh kamera menjadi 320 x 240 piksel.

### 3.5.4 Teknik Socket

Setelah dilakukan penghematan energi dan data, metode penunjang sistem lainnya adalah berupa komunikasi secara client-server antara kedua Raspberry Pi. Komunikasi yang dilakukan disini dengan menggunakan bantuan kabel ethernet RJ45 untuk menghubungkan kedua mikroprosesor tersebut. Berikut ini adalah algoritma socket yang digunakan dalam desain sistem:

```
int main(int argc, char *argv[])
{
```

```

int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;
    char buffer[256];

    pthread_t threadid;
pthread_create(&threadid, NULL, rekam, NULL);

portno = 4545;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0)
    printf("ERROR opening socket\n");

    server = gethostbyname("192.168.1.69");
    if (server == NULL)
    {
        fprintf(stderr, "ERROR, no such host\n");
        exit(0);
    }

    bzero((char *) &serv_addr, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
bcopy((char *)server->h_addr,
    (char *)&serv_addr.sin_addr.s_addr,
    server->h_length);
serv_addr.sin_port = htons(portno);

    if (connect(sockfd,(struct sockaddr *)
&serv_addr,sizeof(serv_addr)) < 0)
        printf("ERROR connecting\n");

while(1)
{
    int _perintah;
    cin >> _perintah;

    bzero(buffer,256);

```

```

        sprintf(buffer, "%d", _perintah);
        n = write(sockfd,buffer,strlen(buffer));

        pthread_mutex_lock(&mutex1);
        perintah = _perintah;
        pthread_mutex_unlock(&mutex1);
        if (perintah == 2)
            break;
    }
}

```

Socket adalah suatu mekanisme komunikasi yang memungkinkan terjadinya pertukaran data antar program atau proses baik dalam satu mesin maupun antar mesin lainnya yang berbeda. Teknik socket menghubungkan kedua mikroprosesor Raspberry Pi dengan kabel ethernet, dimana mekanisme kerja yang dibuat adalah ketika kedua mikroprosesor sudah terhubung menggunakan ethernet kabel, lalu mikroprosesor yang bertindak sebagai *client* mengirimkan data perintah untuk sinkronisasi pengambilan gambar pada masing-masing kamera. Perintah data yang dikirimkan terdapat pada Tabel 3.2.

**Tabel 3.2 Tabel Perintah Data untuk Socket Communication**

| Data yang dikirim dari Client | Data yang diterima oleh server | Perintah yang dilakukan |
|-------------------------------|--------------------------------|-------------------------|
| 1                             | 1                              | Start Recording         |
| 0                             | 0                              | Pause                   |
| 2                             | 2                              | Stop and Close          |

### 3.5.5 Teknik Multithread

Teknik *Multi Thread* pada implementasi platform VSN berfungsi sebagai kontrol pada suatu proses. Implementasi ini menggunakan 2 buah node yang masing-masing node terdiri dari 1 Raspberry Pi yang terdipasang oleh 1 kamera webcam. Antara kedua node tersebut terhubung oleh kabel RJ45 pada port ethernet yang terdapat pada Raspberry Pi.

Multi Thread yang digunakan adalah model Multi Thread One-to-One. Berikut ini adalah algoritma untuk fungsi Multi Thread:

```
pthread_t threadid;
pthread_create(&threadid, NULL, rekam, NULL);

portno = 4545;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0)
    printf("ERROR opening socket\n");

server = gethostbyname("192.168.1.69");
if (server == NULL)
{
    fprintf(stderr, "ERROR, no such host\n");
    exit(0);
}

bzero((char *) &serv_addr, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
bcopy((char *)server->h_addr,
      (char *)&serv_addr.sin_addr.s_addr,
      server->h_length);
serv_addr.sin_port = htons(portno);

if (connect(sockfd,(struct sockaddr *)
&serv_addr,sizeof(serv_addr)) < 0)
    printf("ERROR connecting\n");

while(1)
{
    int _perintah;
    cin >> _perintah;

    bzero(buffer,256);
    sprintf(buffer, "%d", _perintah);
    n = write(sockfd,buffer,strlen(buffer));
```

```

        pthread_mutex_lock(&mutex1);
        perintah = _perintah;
        pthread_mutex_unlock(&mutex1);
    if (perintah == 2)
        break;
    }
}

```

Multi Thread membantu kedua Raspberry Pi yang saling terhubung menggunakan kabel RJ45 melalui port ethernet dengan sistem SSH dapat melakukan pemrosesan data pada mikroprosesor secara paralel. Sehingga waktu perekaman antara kedua kamera pada node yang berbeda tetap dapat merekam dengan waktu yang sama sehingga tidak ada perbedaan antara kedua node tersebut.

### 3.6 Implementasi Sistem *Visual Sensor Networks*

Pada sistem yang dirancang pada tugas akhir ini, bertujuan untuk membuat sistem kerja dari VSN yang dapat menghemat energi dan memori yang disimpan daripada hasil outputan video. Komponen pendukung dari sistem ini antara lain kamera webcam, mikroprosesor, dan hardware pendukung mikroprosesor lainnya. Karena menggunakan 2 buah mikroprosesor maka proses pendeteksian dan perekaman harus dilakukan secara bersamaan dengan sistem SSH dan *socket communication* dengan sistem client-server.

Pada subbab ini akan dibahas bagaimana dari proses pengimplementasian alat serta dokumentasi pada proses pengambilan data dari sistem VSN ini.

#### 3.6.1 Pemanggilan Program pada *Visual Sensor Networks*

Menjalankan sistem penghematan pada VSN ini, ada beberapa tahapan persiapan yang perlu dilakukan. Hal ini dilakukan untuk mempersiapkan sistem kerja agar berjalan dengan baik. Sistem ini, komponen yang perlu dipersiapkan antara lain 2 buah mikroprosesor Raspberry Pi, 2 buah kamera webcam, dan perangkat keras pendukung mikroprosesor lainnya.

Pemanggilan fungsi algoritma yang telah dirancang dan diprogramkan melalui bahasa pemrograman C++ dapat dimunculkan dalam perangkat lunak yang terdapat pada Raspberry Pi bersistem operasi Linux yaitu LXTerminal.

Setelah membuka LXTerminal, selanjutnya dapat lakukan *Change Directory* (cd) untuk melakukan pemanggilan kepada folder tujuan yang dimana didalam folder tersebut terdapat file dari program yang akan dipanggil. Setelah melakukan penggantian directory path, maka selanjutnya kita dapat mengubah IP dan melakukan komunikasi antara kedua mikroprosesor Raspberry Pi untuk saling dapat melakukan proses perekaman secara bersamaan.

**3.6.2 Komunikasi Server-Client Pada Raspberry Pi**

Langkah awal untuk melakukan metode server-client adalah dengan mengubah IP (Internet Protocol) dari kedua Raspberry Pi agar tidak berbenturan karena memiliki IP yang sama. Sebelumnya untuk menghubungkan kedua mikroprosesor Raspberry Pi tersebut, digunakan kabel RJ45 pada ethernet port sebagai *interface* media komunikasinya. Sistem Client-Server digunakan untuk memberikan perintah berupa pengiriman data berupa perintah yang dicerminkan dengan angka 0 1 2 dimana masing-masing angka mempunyai fungsi perintah yang berbeda seperti yang dijelaskan pada subbab 3.5.4 diatas.

**Tabel 3.3 Perubahan IP dan Subnet Mask Pada Raspberry Pi**

|                                 | Raspberry Pi B+<br>(Server) | Raspberry Pi B+<br>(Client) |
|---------------------------------|-----------------------------|-----------------------------|
| <b>IP Default</b>               | 10.122.1.69                 | 10.122.1.69                 |
| <b>IP Sekarang</b>              | 192.168.1.69                | 192.168.1.70                |
| <b>Subnet Mask Default</b>      | 255.255.255.0               | 255.255.255.0               |
| <b>Subnet Mask<br/>Sekarang</b> | 255.255.255.0               | 255.255.255.0               |

Proses perubahan IP masing-masing IP, Raspberry Pi yang bertindak sebagai server menggunakan IP 192.168.1.69 dengan Subnet Mask 255.255.255.0 dan IP dari Raspberry Pi yang bertindak sebagai client menggunakan IP 192.168.1.70 dengan Subnet Mask 255.255.255.0.

Setelah IP kedua Raspberry Pi diubah, selanjutnya menggunakan sistem SSH (*Secure Shell*) agar dapat mengendalikan kedua Raspberry Pi hanya cukup dari salah satu Raspberry Pi nya tersebut. Langkah untuk menggunakan SSH adalah

Ssh -X root@ip

Setelah sistem terhubung secara SSH dan telah dapat berkomunikasi secara client-server, maka langkah selanjutnya adalah dengan memanggil dan mengaktifkan algoritma untuk sistem kerja VSN. Aktifasinya terlebih dahulu kita harus memanggil algoritma pada Raspberry Pi server.

Setelah pemanggilan algoritma pada server, kamera tidak akan langsung merekam, namun kamera akan dalam posisi idle yaitu dimana kamera aktif namun tidak melakukan tindakan apapun. Setelah itu, barulah kita panggil algoritma pada Raspberry Pi client. Seperti halnya pada proses pemanggilan algoritma pada Raspberry Pi server, kamera tidak akan langsung bekerja, namun hanya akan dalam kondisi idle. Untuk mengaktifkan dan menjalankan kamera, client mengirimkan data berupa perintah yang diterima server. Ketika data perintah tersebut dapat terkirim, barulah kedua kamera aktif untuk melakukan perekaman secara real time dan bersamaan.

### **3.6.3 Pendeteksian Objek dan Metode Kompresi Resolusi**

Sesuai dengan tujuan dari tugas akhir ini, yaitu tentang penghematan baik dari segi energi maupun dari segi penghematan memori yang dipakai, maka implementasi pada sistem kerja inipun selaras dengan fungsi tersebut. Apabila pada kamera CCTV (Closed Circuit Television) dan VSN pada umumnya selalu merekam semua kejadian secara utuh, baik ketika daerah pengawasan kamera dalam keadaan ada objek atau tidak ada objek, maka pada sistem kerja VSN disini menggunakan sistem pendeteksian gerakan pada objek yang ditangkap oleh kamera.

Ketika kamera mendeteksi adanya pergerakan dari objek yang ditangkap dalam daerah jangkauannya, maka secara otomatis kamera akan merekam kejadian tersebut. Namun apabila tidak ada objek yang bergerak pada daerah jangkauan kamera tersebut, maka kamera akan dalam posisi idle, tidak merekam namun tidak pula mati. Algoritma yang aktif adalah *Background Substration*.

Selain menggunakan metode pendeteksian pada pergerakan, cara lain yang digunakan untuk mengurangi penggunaan memori yaitu dengan menggunakan metode kompresi resolusi. Kompresi resolusi disini adalah dengan memperkecil resolusi video output yang disimpan kedalam mikroprosesor dibanding dengan resolusi yang ditangkap oleh kamera. Namun pengurangan resolusi disini tetap dengan pertimbangan

yaitu gambar yang dihasilkan tetap dalam resolusi setengah dari resolusi yang ditangkap oleh kamera. Kamera menangkap 640 x 480 piksel maka hasil video akan beresolusi 320 x 240 piksel.

### 3.6.3 Penyimpanan Data Citra Hasil Uji Coba

Setelah proses pengambilan gambar, maka selanjutnya adalah video yang direkam oleh kamera disimpan kedalam Raspberry Pi. Namun dengan keterbatasan kemampuan Raspberry Pi, maka gambar yang disimpan tetap harus tersimpan di masing-masing Raspberry Pi. Jadi gambar yang direkam dalam kamera server, tetap akan tersimpan pada memory di Raspberry Pi server. Begitu juga dengan yang terjadi pada Raspberry Pi client, hasil rekaman akan tersimpan pada memory pada Raspberry Pi client. Proses menjalankan atau menguji coba daripada rekaman yang telah diambil, digunakan fungsi

|                           |
|---------------------------|
| Omxplayer (nama file).avi |
|---------------------------|

### 3.7 Skenario Pengujian

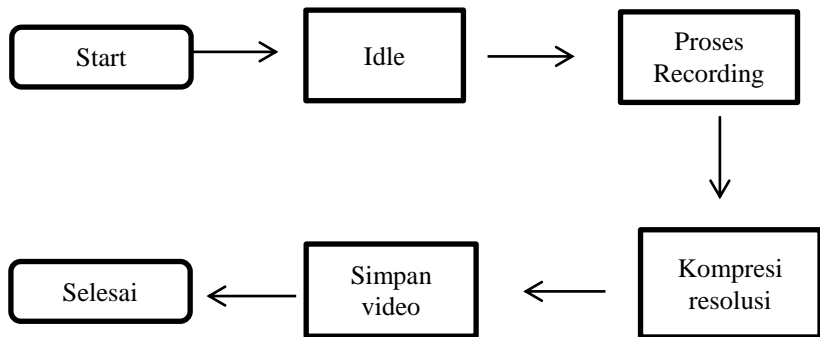
Skenario perekaman disini akan dijelaskan alur sistem kerja yang diawali dari pendeteksian pergerakan, proses pengambilan gambar, dan juga pengolahan video sehingga dihasilkan keluaran video yang diinginkan dan hasil dari sistem kerja *Visual Sensor Network* yang dapat menghemat energi.

#### 3.7.1 Blok Diagram Pengujian

Diagram blok diatas menjelaskan kerja sistem dari perekaman VSN. Apabila sistem tidak terdapat objek yang melintas pada daerah jangkauannya, maka kamera tidak akan melakukan proses perekaman dan kamera akan selalu dalam keadaan *idle*. Ketika terdapat objek yang melintas atau tertangkap kamera, barulah kamera tersebut akan melakukan proses perekaman. Objek yang melintas akan ditangkap oleh kamera dengan resolusi kamera yang ditangkap akan dilakukan kompresi dengan resolusi kamera yang tersimpan oleh mikroprosesor.

Setelah semua proses terlewat, maka hasil video yang ditangkap akan disimpan dalam mikroprosesor Raspberry Pi. Proses ini akan berulang secara terus menerus apabila terdapat objek yang melintas pada daerah jangkauannya. Dan semua proses akan berulang ketika setiap kamera berhasil mendeteksi objek yang bergerak dan melintas pada daerah jangkauan tiap-tiap kamera.





**Gambar 3.5 Blok Diagram Sistem**

### **3.7.1 Parameter Pengujian**

Parameter yang diujikan dalam penelitian ini ada 3, yaitu parameter bit rate tiap video output, energi bit masing-masing node, dan juga konsumsi daya dari masing-masing node yang digunakan dalam sistem kerja.

Tiap parameter memberikan tujuan pembuktian bahwa untuk parameter bit rate untuk mencari kecepatan data dari sistem kerja penghematan ini. Parameter energi bit untuk mengetahui nilai energi tiap satuan bit yang dihasilkan dari sistem kerja ini dan juga parameter daya yang dihasilkan maka akan dilakukan perbandingan antara energi kondisi sistem normal ataupun diberikan program penghematan.

## **3.8 Pengambilan Data**

Pada proses pengambilan data ada beberapa hal yang perlu untuk diperhatikan untuk menunjang nilai keluaran dari hasil analisis data ataupun hasil pendekatan dalam suatu penelitian, termasuk pada penelitian ini. Pengambilan data merupakan langkah penting dalam suatu penelitian, namun data mentah harus melalui suatu prosedur pengolahan yang biasanya didasarkan pada teori penunjang ataupun berdasarkan pada perhitungan rumus yang berkaitan.

### **3.8.1 Metode Pengambilan Data**

Metode pengambilan data pada implementasi sistem VSN adalah dengan mengukur nominal dari masing-masing arus dan tegangan yang dibutuhkan dari tiap node yang digunakan. Pengukuran dilakukan selama 2 menit tiap masing-masing metode. Lamanya

pengukuran 2 menit menunjukkan nilai yang sudah konstan dan dianggap bernilai konstan. Komponen masing-masing node terdiri dari 1 buah Raspberry Pi, 1 buah kamera webcam M-Tech 1,3 Mp, kabel ekstensi USB 2.0 untuk menjangkau kamera lebih jauh dengan sudut pandang yang lebih baik lagi.

Pengukuran nilai dari arus dan tegangan digunakan alat bantu berupa ampere-volt meter. Besarnya ampere-volt meter disini memberikan informasi secara realtime mengenai nilai arus dan tegangan yang digunakan pada masing-masing node. Setelah itu diketahui nilai dari arus dan tegangan yang dipakai. Nilai arus dan tegangan akan diketahui, maka dapat diketahui nilai dari daya yang digunakan oleh masing-masing node.



**Gambar 3.9 Volt-Ampere Meter untuk Pengukuran Arus dan Tegangan**

### 3.8.2 Pengolahan Data

Setelah dilakukannya pengolahan data, maka langkah selanjutnya adalah dengan melakukan pengolahan data. Ada beberapa persamaan rumus yang digunakan dalam pengolahan data pada penelitian ini.

Penggunaan persamaan rumus untuk menentukan pergerakan dalam suatu kondisi algoritma *background subtraction* adalah sebagai berikut,

$$R(r, c) = I(r, c) - B(r, c) \quad \dots\dots(1)$$

Dengan:

$R(r,c)$  = Nilai *Background Substraction*

$I(r,c)$  = Matriks nilai citra saat ini (*foreground*)

$C(r,c)$  = Matriks nilai citra backgrounds model (*background*)

$r$  = Baris

$c$  = Kolom

Selanjutnya adalah untuk menentukan daya yang digunakan pada masing-masing node. Penentuan besarnya jumlah daya yang dipakai digunakan persamaan rumus sebagai berikut

$$P = V \times I \quad \text{.....(2)}$$

Dengan:

$P$  = Besarnya nilai daya yang dikonsumsi (Watt)

$I$  = Nilai besaran arus pada suatu node (A)

$V$  = Nilai tegangan pada suatu node (volt)

Bit rate merupakan parameter umum yang biasa digunakan dalam suatu file baik gambar, suara, maupun video. Bit rate nantinya dapat digunakan untuk melakukan perhitungan dalam menentukan besarnya nilai dari energi bit yang dikonsumsi oleh sistem kerja VSN. Salah satu cara untuk menentukan besarnya nilai bit rate adalah dengan mengubah ukuran video menjadi satuan kilobit dan membaginya dengan durasi lamanya video tersebut.

$$Bit Rate = \frac{Ukuran Video \times 8 \times 1000}{Durasi Video} \quad \text{.....(3)}$$

Sementara untuk menentukan energi dari masing-masing bit yang dihasilkan dari hasil perekaman dengan menggunakan persamaan  $E_b/N_o$ . Persamaan tersebut merupakan parameter umum yang digunakan untuk membandingkan sistem komunikasi bahkan jika terdapat perbedaan bit rate, modulasi, dan bahkan media. Kuantitas  $E_b$  adalah ukuran dari *Bit Energy*.

$$E_b = \frac{P_{avg}}{R_b} \quad \text{.....(4)}$$

Dimana:

$P_{avg}$  = Level Daya Rata-rata

$R_b$  = Bit Rate

## **BAB 4**

### **ANALISIS DATA**

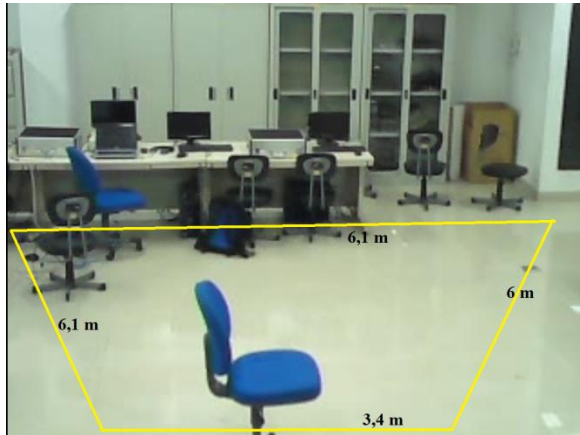
Pada bab ini akan dibahas mengenai analisis dari sistem kerja VSN yang dibuat. Sistem perekaman pada *Visual Sensor Networks* yang menggunakan metode pendeteksian pergerakan objek. Pada sistem kerja yang telah dibuat pada tugas akhir ini akan dilakukan pembahasan dan analisis data untuk mengetahui performa kerja dan kondisi pengaplikasian kerja terbaiknya. Berikut beberapa pembahasan analisis yang terdiri dari analisis kerja algoritma simulasi sederhana, analisis kerja algoritma implementasi, pengaruh konsumsi daya, kebutuhan energi bit rate, dan pengaruh kompresi resolusi terhadap kinerja dan hasil kerja.

#### **4.1 Analisis Algoritma *Visual Sensor Networks***

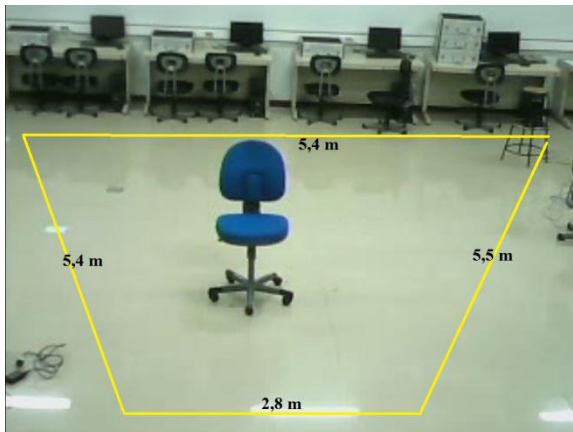
Seperti yang telah dibahas sebelumnya, bahwa teknologi dari *Wireless Sensor Networks* ataupun *Visual Sensor Networks* memiliki perkembangan teknologi yang sangat pesat, namun tetap ada kendala dalam pengimplementasiannya. Salah satunya adalah masalah energi yang diperlukan. Subbab ini akan dijelaskan tentang metode penghematan dari sistem kerja *Visual Sensor Networks*. Subbab ini pula akan dibahas mengenai analisa kerja dari metode algoritma uji coba yang dirancang untuk menghemat energi.

##### **4.1.1 Skema Ruangan dan Daerah *Overlap***

Seperti yang telah dijelaskan pada bab sebelumnya, ruangan yang digunakan pada pengimplementasian ini adalah Ruang Rangkaian Digital AJ403 Jurusan Teknik Elektro ITS. Tiap-tiap node yang digunakan mempunyai daerah *coverage* atau jangkauan masing-masing. Jangkauan itu didapatkan nilai dari besarnya daerah *overlap* atau daerah yang beririsan antara kedua node kamera tersebut. Pada wilayah *overlap* tersebut, kondisi dari kedua node akan saling merekam. Berbeda dengan kondisi diluar daerah *overlap* kedua node, dalam kondisi tersebut hanya node dengan kamera yang menangkap pergerakan objek sajalah yang akan melakukan perekaman.



**Gambar 4.1 Daerah Jangkauan Node Client**



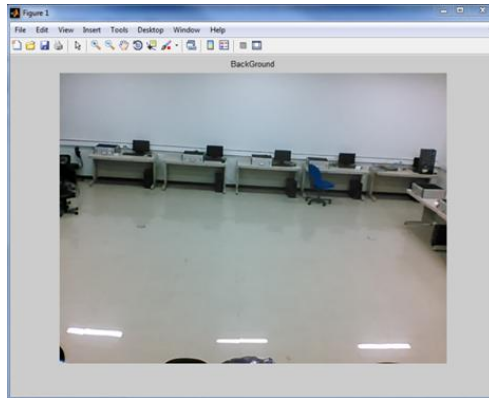
**Gambar 4.2 Daerah Jangkauan Node Server**

Kedua gambar diatas dapat kita lihat bersama bahwa daerah jangkauan kedua kamera memiliki bentuk trapesium yang tidak simetris. Kedua daerah jangkauan kamera tersebut memiliki luas daerah overlap masing-masing.

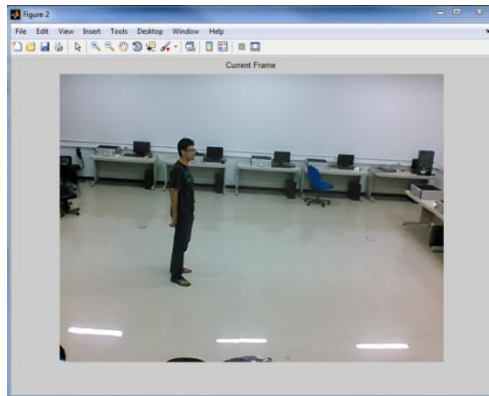
#### **4.1.2 Analisis Sistem Simulasi Sederhana**

Pengujian ini, variabel yang dianalisis adalah kemampuan deteksi objek pada daerah jangkauan masing-masing kamera. Dengan menggunakan kemampuan algoritma *Background Subtraction* maka objek yang bergerak dapat dideteksi. Fungsinya adalah dengan

mengambil frame suatu kejadian yang ditangkap oleh kamera, lalu mengurangkan matriksnya dengan frame selanjutnya. Selanjutnya hasil dari pengurangan dua matriks frame tadi akan menuju pada sebuah nilai hasil pengurangan matriks. Nilai pengurangan matriks tersebut nantinya yang akan memberikan informasi, apakah dalam suatu frame yang ditangkap tersebut ada sebuah pergerakan atau tidak melalui margin jangkauan *threshold*.



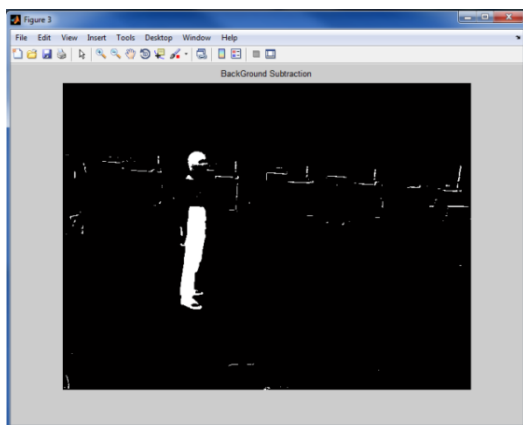
**Gambar 4.3 Frame Referensi Ruangan Kosong**



**Gambar 4.4 Frame Referensi dengan Objek**

2 gambar yang terdapat diatas didapatkan bahwa kondisi tersebut terjadi pergerakan. Seperti gambar yang ada, Gambar 4.3 merupakan frame ruangan kosong yang dimana frame ini dijadikan *background frame* sedangkan pada Gambar 4.4 terdapat objek yang tertangkap dalam frame yang diambil.

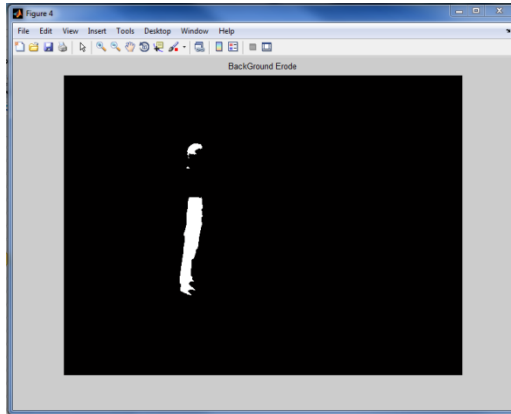
Pergerakan objek yang ada dalam frame tidak terlihat dengan jelas, maka inputan gambar selanjutnya dibuat *grayscale*. Karena dengan menggunakan model gambar *grayscale*, lebih mudah membedakan *background* dan *foreground* dari perubahan gambar. Selain itu dengan model gambar *grayscale*, akan lebih sedikit noise yang terdapat pada gambar.



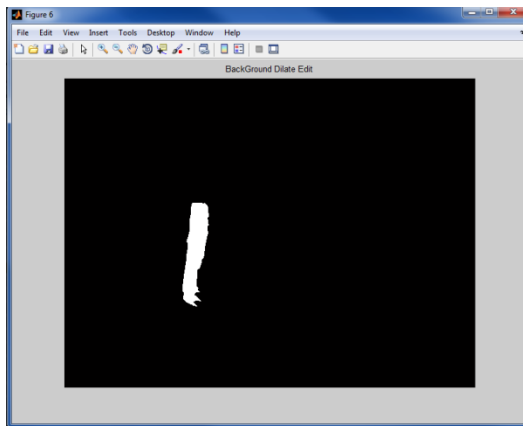
**Gambar 4.5 Hasil Output *Grayscale***

Namun gambar diatas masih terdapat banyak *noise*, yang disimpulkan dengan banyaknya notasi putih dari foreground. Untuk mengatasinya dilakukan pengubahan gambar dengan menggunakan *dilate* dan *erode*. Fungsi dilakukannya dilate disini adalah untuk mengisi bagian dari frame yang kosong, dan fungsi dari erode adalah menghapus pixel noise yang berwarna putih tersebut. Setelah dilakukan proses *dilate* dan *erode* maka akan dihasilkan gambar yang hanya menunjukkan bentuk pergerakan objek benda, yang di dapatkan dari hasil pengurangan matriks frame foreground terhadap matriks frame background.

Menurut persamaan (1) pada bab 3, maka teknik pendeteksian pada simulasi akan terlihat seperti berikut,



**Gambar 4.6 Hasil Citra Setelah Proses *Erode Image***



**Gambar 4.7 Hasil Citra Setelah Proses *Dilate Image***

Selanjutnya untuk menentukan keberadaan objek, apakah dalam algoritma simulasi sederhana tersebut mendeteksi keberadaan objek yang bergerak atau tidak. Setelah menjalankan program simulasi sederhana dengan menggunakan Matlab, maka akan terlihat pada



*command window* bahwa akan ada informasi yang akan diberikan oleh fungsi Matlab tersebut.

```
Command Window

Keliling blob 1 adalah 367 pixel
ADA PERGERAKAN
Keliling blob 2 adalah 18 pixel
Keliling blob 3 adalah 77 pixel
Keliling blob 4 adalah 13 pixel
```

**Gambar 4.8 Deteksi Pergerakan Objek Pada Simulasi Sederhana**

Gambar di atas dapat disimpulkan bahwa pada pendeteksian objek menggunakan simulasi sederhana Matlab dapat mendeteksi adanya pergerakan dari objek. Seperti informasi dari keluaran yang diberikan oleh Matlab tersebut bahwa “ADA PERGERAKAN”. Dari gambar diatas juga diketahui bahwa keliling blob untuk gambar 1 (*foreground*) adalah 367 piksel. Sedangkan untuk gambar selanjutnya setelah dilakukan *dilate* dan *erode* sebesar 18 piksel, 77 piksel, dan 13 piksel. Sementara itu, untuk mengetahui koordinat dari objek bisa menggunakan algoritma yang dapat menentukan titik atau pixel paling bawah. Yaitu dengan pemanggilan algoritma pada Matlab *command window*.

```
m =

    480

>> n

n =

    640
```

**Gambar 4.9 Deteksi Koordinat Pixel**

Gambar di atas didapatkan bahwa koordinat daripada pixel letak objek adalah pada pixel dengan tinggi 640 pixel dan lebar 480 pixel. Simbol “m” mewakili daripada panjang dari pixel letak objek, dan

simbol “n” mewakili dari lebar. Hal ini dapat menentukan koordinat objek dalam bentuk pixel agar mendapatkan informasi lokasi dari objek yang bergerak.

Pengambilan data dari simulasi sederhana disini, tidak hanya mendeteksi ada atau tidaknya pergerakan dari objek, tetapi juga menentukan koordinat daripada objek yang terbaca pergerakannya. Melalui pixel yang dimana objek itu berdiri, kita dapat menentukan koordinatnya letak dari objek yang bergerak tersebut sesuai dengan data diatas.

#### **4.1.3 Analisis Sistem Implementasi *Visual Sensor Networks***

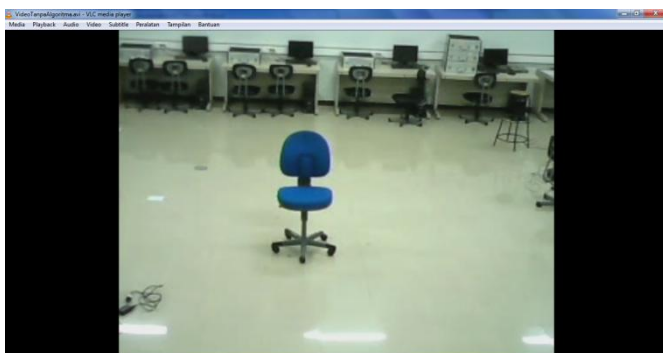
Pengimplementasian dari VSN dengan menggunakan mikroprosesor Raspberry Pi dan kamera webcam dengan resolusi 1,3 Mp digunakan sistem SSH dalam pengukuran dan pengambilan data karena keterbatasan infrastruktur. Menggunakan LXTerminal dan mengganti IP kedua Raspberry Pi yang digunakan dan membuat sistem server-client dimana IP server adalah 192.168.1.69 dan IP dari client adalah 192.168.1.69. Ketika kedua Raspberry Pi telah saling berkomunikasi dengan menggunakan algoritma socket, maka kedua Raspberry Pi dapat saling mengirim perintah data. Seperti yang dijelaskan sebelumnya, pengiriman data berupa perintah untuk dapat menjalankan program dalam bentuk algoritma pada LXTerminal. Menggunakan 2 kamera, dimana 1 bertugas sebagai client dan 1 lainnya sebagai server, maka pengambilan gambar atau perekaman akan menggunakan sistem shifting

##### **4.1.2.1 Pengambilan VideoTanpa Algoritma**

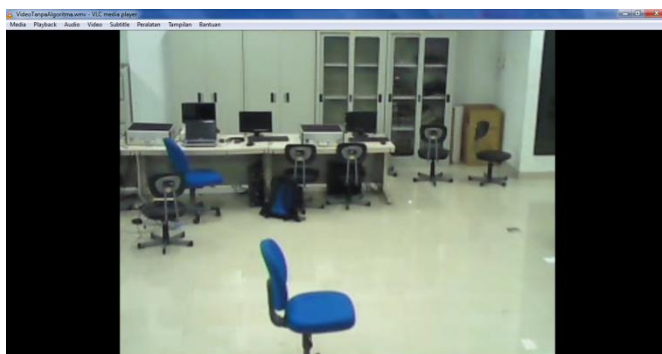
Dilakukan proses pengambilan gambar dengan sistem normal tanpa algoritma digunakan sebagai pembanding untuk menguji efektifitas dan kelayakan guna dari algoritma yang dirancang dalam penelitian ini. Kondisi normal tidak dilakukan pendeteksian sama sekali terhadap pergerakan dalam daerah cakupannya. Hal ini biasanya diterapkan dalam kamera pengawas pada umumnya.

Gambar di bawah adalah gambar hasil *capture* dari pengambilan gambar dengan sistem kerja normal atau tanpa diberikan algoritma efisien. Gambar di bawah didapatkan bahwa kamera merekam keseluruhan kejadian pada lingkungan kerja atau daerah coverage dari masing-masing node. Kondisi ruangan tersebut terdapat objek berupa kursi, meja dan beberapa perangkat praktikum seperti komputer, CPU,

dan lainnya. Namun karena pengambilan data disini adalah untuk mendeteksi pergerakan dari suatu objek dalam hal ini adalah objek gerakan manusia, maka tidak terdeteksi apapun dalam gambar tersebut. Objek benda mati lainnya tidak termasuk kedalam pergerakan karena tidak bergerak atau diam dan tidak termasuk kedalam threshold yang telah ditentukan sebagai parameter pergerakan. Mode perekaman normal ini banyak diaplikasikan pada CCTV atau media perekaman lainnya.



**Gambar 4.10 Gambar Video Tanpa Algoritma Node Server**



**Gambar 4.11 Gambar Video Tanpa Algoritma Node Client**

Berdasarkan cuplikan gambar diatas, dapat kita lihat bersama bahwa ketika suatu daerah jangkauan tidak terdapat pergerakan sekalipun, maka akan tetap dilakukan proses perekaman gambar. Berdasarkan algoritma yang dibuatnya, tidak tergantung pada suatu

pererakan sekalipun maka akan selalu dilakukan perekaman gambar pada daerah jangkauannya.

#### ***4.1.2.2 Pengambilan Video dengan Algoritma***

Selanjutnya adalah analisa hasil perekaman menggunakan metode algoritma yang dirancang pada penelitian ini. Masih dengan sistem perangkat kerja yang sama dengan 2 node dimana masih ada pembagian peran baik node sebagai client dan sebagai server. Perbedaan antara keduanya adalah proses dalam merekam. Algoritma ini akan bekerja merekam kejadian yang ditangkap dalam daerah *coverage* dengan pendeteksian pergerakan objek. Berikut adalah cuplikan gambar dari pengambilan gambar dengan video algoritma.



**Gambar 4.12 Gambar Video dengan Algoritma Node Server**



**Gambar 4.13 Gambar Video dengan Algoritma Node Client**

Berdasarkan gambar diatas dapat kita lihat bersama, ketika ada pergerakan objek yang masuk kedalam lingkup *coverage* maka akan melakukan perekaman terhadap objek benda yang terlintas dalam daerah *coverage*. Ketika dalam suatu frame tidak terdeteksi suatu pergerakan, maka sesuai perintah dari algoritma maka sistem kerja ini tidak melakukan proses perekaman gambar. Ketika tidak adanya pergerakan dari objek yang melintas pada lintasan daerah jangkauan, maka node akan dalam kondisi *idle* tidak melakukan proses perekaman hanya dalam stand by mode.

#### **4.2 Analisis Pengaruh Konsumsi Daya**

Seperti yang telah dijelaskan pada bab sebelumnya, konsumsi energi merupakan salah satu masalah yang memang menjadi halangan dalam perkembangan jenis teknologi ini. Aplikasi VSN pada umumnya, dengan selalu melakukan proses *recording* pada lingkungan yang diamati oleh kamera, membuat konsumsi energi menjadi lebih boros. Apabila lingkungan yang diamati oleh kamera tidak terjadi apapun, akan tetapi apabila di manajemen dengan hanya melakukan proses pengambilan gambar ketika terdapat objek.

Diterapkan sistem perekaman kejadian yang berdasarkan dengan deteksi objek bergerak, akan dilihat pengaruh daya yang dikonsumsi antara sistem kerja VSN dengan algoritma dan sistem kerja VSN tanpa algoritma. Akan dilihat perbandingan keduanya dengan menggunakan tipe kamera yang berbeda, yaitu kamera yang bertugas sebagai client dan server.

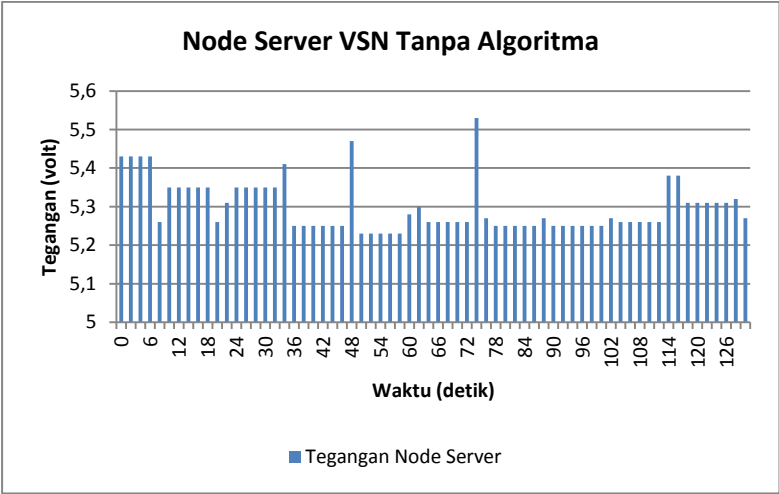
Pengukuran konsumsi daya dilakukan dengan melakukan perekaman dengan menggunakan sistem kerja VSN algoritma dan tanpa algoritma. Pengukuran dilakukan selama 2 menit, setelah itu diambil data daya yang dibutuhkan. Persamaan (2) pada bab 3 makan, didapatkan dari pengukuran nilai besaran arus dan tegangan. Dari hasil pengukuran tersebut dapat dilakukan perhitungan terhadap besarnya nilai daya yang dikonsumsi oleh tiap node, dimana tiap node terdiri dari kamera dan mikroprosesor Raspberry Pi.

Pengukuran arus dan tegangan dilakukan dengan bantuan alat volt-ampere meter. Secara langsung terukur besarnya arus dan tegangan yang digunakan oleh masing-masing node tersebut. Hasil pengukuran sistem VSN yang telah dilakukan, didapatkan tegangan dan arus yang diambil. Pendataan arus dan tegangan diambil setiap 2 detik selama 2 menit. Sehingga didapatkan data arus dan tegangan rata-rata

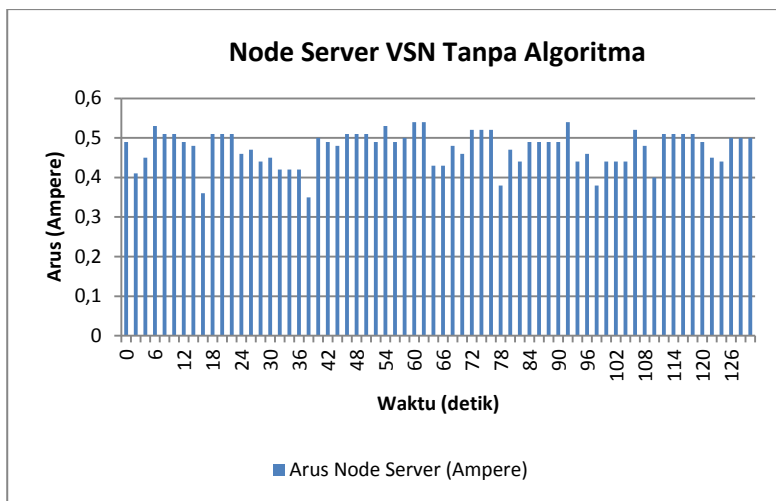
tiap node server dan node client, selanjutnya menghitung daya rata-rata yang dikonsumsi oleh masing-masing node.

**4.2.1 Daya Tanpa Penggunaan Algoritma**

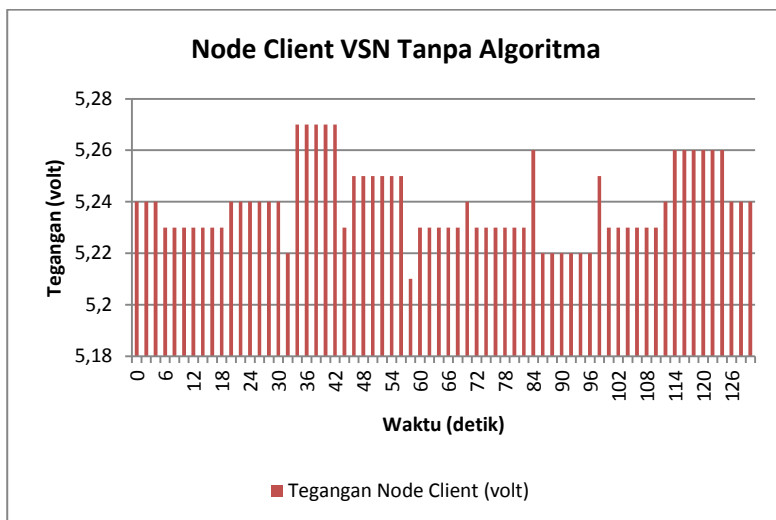
Dalam pengujian ini, variabel yang dianalisis adalah kemampuan deteksi objek. Selain itu hal yang dijadikan parameter adalah konsumsi daya dari masing-masing node. Berikut ini adalah hasil pengukuran yang diambil pada pengukuran VSN dengan tanpa menggunakan algoritma efisien atau kondisi normal.



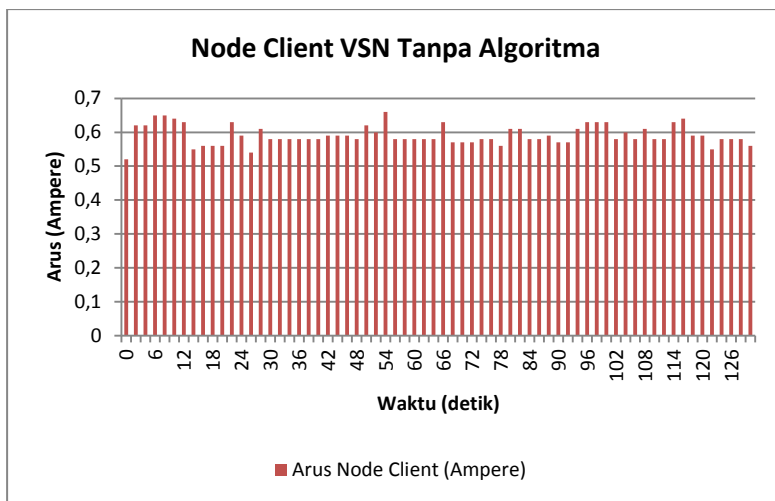
**Gambar 4.14 Grafik Tegangan Node Server Tanpa Algoritma**



**Gambar 4.15 Grafik Arus Node Server Tanpa Algoritma**



**Gambar 4.16 Grafik Tegangan Node Client Tanpa Algoritma**



**Gambar 4.17 Grafik Arus Node Client Tanpa Algoritma**

Didapatkan nilai besaran arus rata-rata dan tegangan rata-rata yang diambil pada pengukuran. Menggunakan rumus perhitungan daya didapatkan, maka didapatkan:

**Tabel 4.1 Tabel Pengukuran Data Arus Tanpa Algoritma**

| Keterangan         | Node Server | Node Client |
|--------------------|-------------|-------------|
| Arus Maksimal (A)  | 0,540       | 0,680       |
| Arus Minimal (A)   | 0,350       | 0,550       |
| Arus Rata-rata (A) | 0,475       | 0,586       |

**Tabel 4.2 Tabel Pengukuran Data Tegangan Tanpa Algoritma**

| Keterangan             | Node Server | Node Client |
|------------------------|-------------|-------------|
| Tegangan Maksimal (v)  | 5,530       | 5,26        |
| Tegangan Minimal (v)   | 5,230       | 5,22        |
| Tegangan Rata-rata (v) | 5,297       | 5,24        |

Data tabel diatas, didapatkan bahwa tegangan rata-rata dari node server adalah sebesar 5,297 V sedangkan nilai dari tegangan rata-rata node client adalah sebesar 5,24 V. Sedangkan untuk arus rata-rata



dari node server adalah sebesar 0,475 A sedangkan nilai arus rata-rata dari node client adalah sebesar 0,586 A.

Didapatkan besaran daya pada masing-masing node pada sistem *Visual Sensor Networks* tanpa algoritma efisien adalah sebagai berikut

**Tabel 4.3 Tabel Pengukuran Data Daya Tanpa Algoritma**

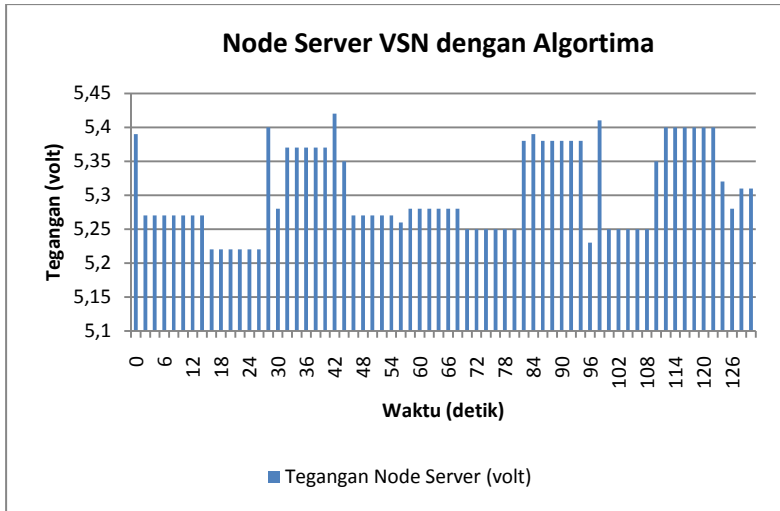
| Keterangan            | Node Server | Node Client |
|-----------------------|-------------|-------------|
| Daya Maksimal (Watt)  | 2,878       | 3,465       |
| Daya Minimal (Watt)   | 1,840       | 2,830       |
| Daya Rata-rata (Watt) | 2,515       | 3,104       |

Didapatkan nilai daya rata-rata dari pengukuran untuk masing-masing node. Node server mempunyai nilai rata-rata sebesar 2,515 Watt dan daya rata-rata pada node client sebesar 3,104 Watt. Nilai rata-rata daya didapatkan dari perkalian antara rata-rata arus dan rata-rata tegangan tiap node. Selama pengukuran 2 menit, didapatkan daya pada masing-masing node adalah seperti tabel diatas.

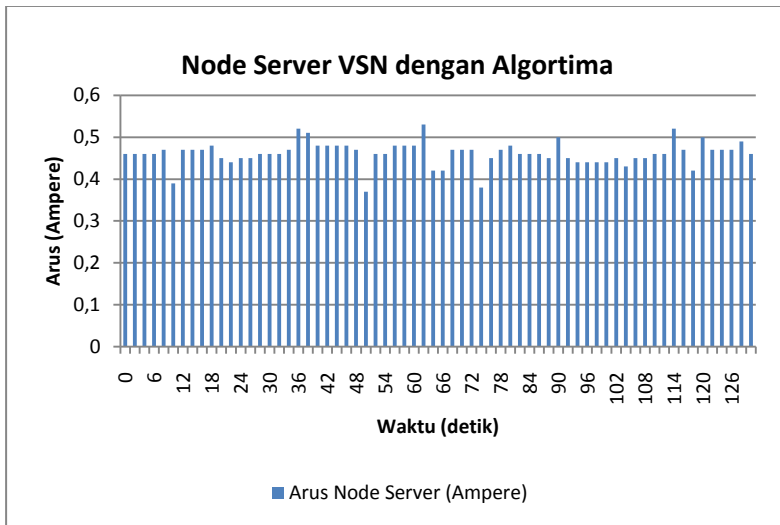
#### **4.2.2 Daya dengan Algoritma**

Setelah mendapatkan nilai dari pengukuran VSN tanpa algoritma efisien atau dalam kondisi normal, maka selanjutnya pengukuran daya pada masing-masing node dengan menggunakan algoritma.

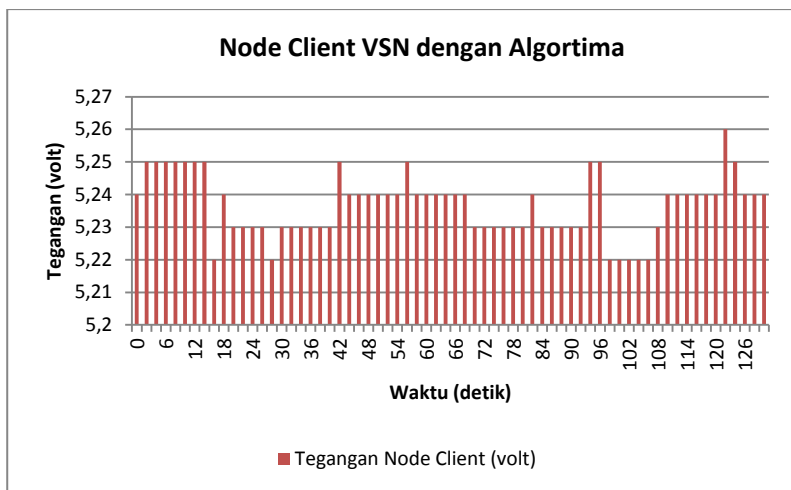
Sama seperti pengukuran pada VSN tanpa algoritma, pengukuran yang akan diambil datanya adalah daya yang dipakai oleh masing-masing node sensor. Data dari tegangan dan daya yang diambil, selanjutnya mencari arus rata-rata dan tegangan rata-rata. Setelah itu kita dapat mengetahui nilai dari daya yang digunakan oleh masing-masing node.



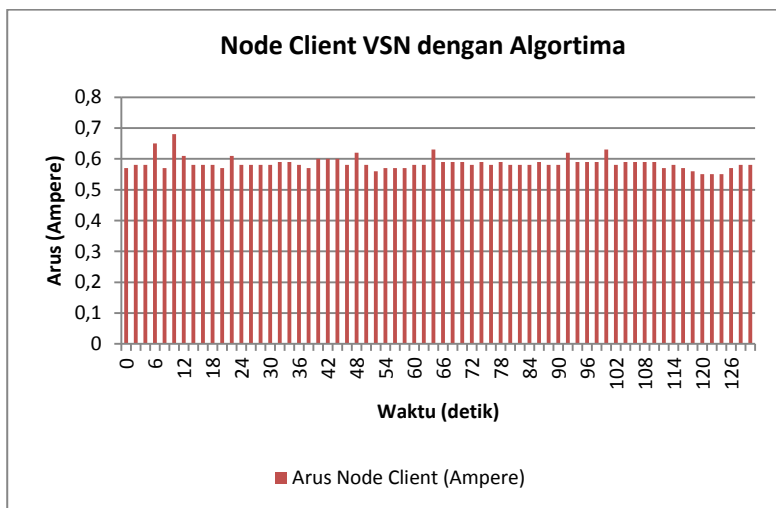
**Gambar 4.18 Grafik Tegangan Node Server dengan Algoritma**



**Gambar 4.19 Grafik Arus Node Server dengan Algoritma**



**Gambar 4.20 Grafik Tegangan Node Client dengan Algoritma**



**Gambar 4.21 Grafik Arus Node Client dengan Algoritma**

Didapatkan nilai besaran arus rata-rata dan tegangan rata-rata yang diambil pada pengukuran. Menggunakan rumus perhitungan daya didapatkan, maka didapatkan:

**Tabel 4.4 Tabel Pengukuran Data Arus dengan Algoritma**

| Keterangan         | Node Server | Node Client |
|--------------------|-------------|-------------|
| Arus Maksimal (A)  | 0,530       | 0,660       |
| Arus Minimal (A)   | 0,370       | 0,540       |
| Arus Rata-rata (A) | 0,461       | 0,592       |

**Tabel 4.5 Tabel Pengukuran Data Tegangan dengan Algoritma**

| Keterangan             | Node Server | Node Client |
|------------------------|-------------|-------------|
| Tegangan Maksimal (v)  | 5,420       | 5,270       |
| Tegangan Minimal (v)   | 5,220       | 5,210       |
| Tegangan Rata-rata (v) | 5,305       | 5,239       |

Didapatkan bahwa tegangan rata-rata dari node server adalah sebesar 5,305 volt sedangkan nilai dari tegangan rata-rata node client adalah sebesar 5,239 volt. Untuk arus rata-rata dari node server adalah sebesar 0,461 A sedangkan nilai arus rata-rata dari node client adalah sebesar 0,592 A.

Didapatkan nilai rata-rata arus dan rata-rata tegangan, maka dari data tersebut dapat ditentukan nilai daya masing-masing node. Setelah mendapatkan data tersebut, nilai dari daya dapat ditentukan.

**Tabel 4.6 Tabel Pengukuran Data Daya dengan Algoritma**

| Keterangan            | Node Server | Node Client |
|-----------------------|-------------|-------------|
| Daya Maksimal (Watt)  | 2,808       | 3,570       |
| Daya Minimal (Watt)   | 1,950       | 2,882       |
| Daya Rata-rata (Watt) | 2,448       | 3,069       |

Didapatkan nilai daya rata-rata dari pengukuran untuk masing-masing node. Node server mempunyai nilai rata-rata sebesar 2,448 Watt dan daya rata-rata pada node client sebesar 3,069 Watt. Nilai rata-rata daya didapatkan dari perkalian antara rata-rata arus dan rata-rata tegangan tiap node. Selama pengukuran 2 menit, didapatkan daya pada masing-masing node adalah seperti tabel diatas.

### 4.2.3 Perbandingan Daya

Setelah mendapatkan kedua nilai daya rata-rata yang digunakan oleh masing-masing node dengan 2 macam algoritma yang berbeda. Dengan algoritma efisien dan tanpa algoritma efisien atau kondisi normal.

**Tabel 4.7 Tabel Perbandingan Daya Rata-rata**

| Node        | Rata-rata Daya Tanpa Algoritma (Watt) | Rata-rata Daya dengan Algoritma (Watt) |
|-------------|---------------------------------------|--|
| Node Server | 2,515                                 | 3,104                                  |
| Node Client | 2,448                                 | 3,069                                  |
| Total Daya  | 4,963                                 | 6,173                                  |

Dengan demikian, berdasarkan data yang telah diambil, kita dapat menentukan persentase penghematan daya yang digunakan oleh metode penghematan.

$$\begin{aligned}
 & \text{Penghematan \%} \\
 &= \frac{\text{Selisih Daya Rata - rata}}{\text{Jumlah Rata - rata Daya Kondisi Normal}} \times 100\% \\
 &= \frac{(6,173-4,963)}{6,173} \times 100\% \\
 &= \frac{1,210}{6,173} \times 100\% \\
 &= 19,6 \%
 \end{aligned}$$

Maka setelah dilakukan perbandingan, dapat disimpulkan bahwa dengan metode ini dapat menghemat daya sebesar 19,6%. Hal ini dikarenakan pada metode ini, tidak selalu kamera tersebut melakukan perekaman terhadap objek. Kamera pada node hanya akan melakukan perekaman apabila dalam frame ada objek yang terdeteksi. Sehingga memungkinkan kamera melakukan penyimpanan daya dalam posisi *idle*.

### 4.3 Analisis Energi Bit Rate

Parameter selanjutnya adalah perbandingan bit energi yang digunakan dalam proses ini. Terjadi perbedaan bit rate antara kedua video hasil output kedua jenis metode. Baik sistem VSN yang menggunakan algoritma maupun sistem kerja yang tidak menggunakan algoritma efisien atau kondisi normal.

Seperti pengertian bit rate, bahwa bit rate merupakan suatu ukuran kecepatan bit suatu data dari tempat satu ke tempat lain yang biasanya diukur dengan waktu seperti Kbps (Kilobit per second), Mbps (Megabit per second) dan lainnya. Bitrate disini digunakan tidak hanya untuk mengetahui kecepatan data, namun juga dapat menghitung bit energi dan juga mengetahui ukuran daripada video yang disimpan dari proses perekaman. Hal ini dapat menjadi tolak ukur untuk mengetahui penghematan dari metode algoritma efisiensi.

#### 4.3.1 Analisis Ukuran Video

Selain dari segi penghematan energi bit, metode algoritma efisien ini juga dapat menghemat kebutuhan dari penyimpanan memori. Masing-masing dari video output menghasilkan data sebagai berikut

**Tabel 4.8 Tabel Data Frame Rate Video**

| Parameter   | Tanpa Algoritma |          | Dengan Algoritma |         |
|-------------|-----------------|----------|------------------|---------|
|             | Server          | Client   | Server           | Client  |
| Frame Size  | 320x240         | 320x240  | 320x240          | 320x240 |
| Frame Rate  | 15 fps          | 15 fps   | 15 fps           | 15 fps  |
| Color Depth | 8 bit           | 8 bit    | 8 bit            | 8 bit   |
| Time        | 56 detik        | 62 detik | 8 detik          | 6 detik |

Selain itu didapatkan pula data dari proses pengambilan gambar bahwa terdapat perbedaan dari besarnya file dari masing-masing node dengan kedua metode baik dengan menggunakan algoritma efisien maupun tanpa menggunakan algoritma efisien atau kondisi normal.

**Tabel 4.9 Perbandingan Hasil Pengukuran dan Perhitungan Ukuran File**

| Pengambilan Data | Node        | Hasil Pengukuran |
|------------------|-------------|------------------|
| Dengan Algoritma | Node Server | 1,01 MB          |
|                  | Node Client | 1,65 MB          |
| Tanpa Algoritma  | Node Server | 9,65 MB          |
|                  | Node Client | 12,7 MB          |

Perbandingan yang didapat diatas, maka dapat dianalisis bahwa antara hasil rekaman VSN yang menggunakan algoritma efisien akan memiliki file output yang lebih kecil dibandingkan dengan

menggunakan metode pengambilan gambar tanpa algoritma efisien atau kondisi normal. Hal ini dikarenakan pada kedua metode tersebut, hasil video dengan menggunakan algoritma efisien membutuhkan waktu perekaman lebih lama dibandingkan dengan metode tanpa algoritma. Pada VSN menggunakan algoritma hanya akan merekam apabila terdapat objek didalamnya selain itu node sensor akan berada dalam kondisi *idle*. Berbeda dengan kondisi pada metode normal yang tanpa menggunakan algoritma, node sensor akan selalu melakukan perekaman baik kondisi frame terdeteksi pergerakan maupun tidak terdeteksi pergerakan.

### 4.3.2 Analisa Perhitungan Bit Rate

Sesuai dengan pemaparan persamaan (3) mengenai penghitungan jumlah bit rate, maka sesuai dengan persamaan didapatkan bit rate sesuai dengan teori

#### 4.3.2.1 Bit Rate Video Tanpa Menggunakan Algoritma

- Node Server

$$\text{Bit Rate} = \frac{9,65 \text{ MB} \times 8 \times 1000}{56 \text{ detik}}$$

$$\text{Bit Rate} = 1378,6 \text{ Kbps}$$

- Node Client

$$\text{Bit Rate} = \frac{12,7 \text{ MB} \times 8 \times 1000}{62 \text{ detik}}$$

$$\text{Bit Rate} = 1638,7 \text{ Kbps}$$

#### 4.3.2.2 Bit Rate Video dengan Menggunakan Algoritma

- Node Server

$$\text{Bit Rate} = \frac{1,01 \text{ MB} \times 8 \times 1000}{5 \text{ detik}}$$

$$\text{Bit Rate} = 1616 \text{ Kbps}$$

- Node Client

$$\text{Bit Rate} = \frac{1,65 \text{ MB} \times 8 \times 1000}{7 \text{ detik}}$$

$$\text{Bit Rate} = 1885,7 \text{ Kbps}$$

#### 4.3.2.3 Perbandingan Bit Rate Hasil Perekaman Gambar

Hasil perhitungan bit rate sesuai dengan persamaan pencarian bit rate, didapatkan hasil bit rate seperti tabel dibawah ini:

**Tabel 4.10 Perbandingan Bit Rate Kedua Metode Pengambilan Gambar**

| Bit Rate                  | Node Server | Node Client |
|---------------------------|-------------|-------------|
| Bit Rate Tanpa Algoritma  | 1378,6 Kbps | 1638,7 Kbps |
| Bit Rate dengan Algoritma | 1616 Kbps   | 1885,7 Kbps |

Didapatkan nilai bit rate dari sistem kerja *Visual Sensor Networks* dengan menggunakan algoritma efisien untuk node sensor dan node client. Didapatkan nilai bit rate perbandingan antara algoritma efisien dengan sistem kerja tidak menggunakan algoritma efisien, nilai dari algoritma efisien lebih kecil dibandingkan dengan yang tidak menggunakan algoritma.

#### 4.3.3 Analisa Kebutuhan Bit Energi

Ada beberapa langkah untuk menganalisis energi bit dari video hasil rekaman kejadian, dapat menggunakan persamaan (4) pada bab 3 diatas. Sedangkan *Pavg* dianggap sama untuk mikroprosesor Raspberry Pi yaitu sebesar 3 W atau 3000 mW.

##### 4.3.3.1 Perhitungan Kebutuhan Bit Energi Tanpa Algoritma

Nilai dari energi bit dari perhitungan rumus sebesar

- Node server tanpa algoritma

$$Eb = \frac{3000 \text{ mW}}{1378,6 \text{ Kbps}}$$

$$A = 10 \log 3000 - 10 \log 1378,6$$

$$A = 34,77 - 31,39$$

$$= 3,38 \text{ dBm}$$

$$= 2,18 \text{ mW}$$



- Node client tanpa algoritma

$$Eb = \frac{3000 \text{ mW}}{1616 \text{ Kbps}}$$

$$A = 10 \log 3000 - 10 \log 1638,7$$

$$A = 34,77 - 32,14$$

$$= 2,63 \text{ dBm}$$

$$= 1,83 \text{ mW}$$

#### 4.3.3.2 Perhitungan Kebutuhan Bit Energi dengan Algoritma

Nilai dari energi bit dari perhitungan rumus sebesar

- Node server dengan algoritma

$$Eb = \frac{3000 \text{ mW}}{1616 \text{ Kbps}}$$

$$A = 10 \log 3000 - 10 \log 1616$$

$$A = 34,77 - 32,08$$

$$= 2,69 \text{ dBm}$$

$$= 1,86 \text{ mW}$$

- Node client dengan algoritma

$$Eb = \frac{3000 \text{ mW}}{1885,7 \text{ Kbps}}$$

$$A = 10 \log 3000 - 10 \log 1885,7$$

$$A = 34,77 - 32,75$$

$$= 2,02 \text{ dBm}$$

$$= 1,6 \text{ mW}$$

#### 4.3.3.3 Perbandingan Hasil Pengukuran Energi Bit

Menurut hasil perhitungan jumlah energi bit dari metode pengukuran dan hasil perhitungan, didapatkan masing-masing nilai energi bit sebagai berikut

**Tabel 4.11 Perbandingan Bit Energi**

|       | Tanpa Algoritma |             | Dengan Algoritma |             |
|-------|-----------------|-------------|------------------|-------------|
|       | Node Server     | Node Client | Node Server      | Node Client |
|       | 2,18 mW         | 1,83 mW     | 1,86 mW          | 1,6 mW      |
| Total | 4,01 mW         |             | 3,46 mW          |             |

Berdasarkan nilai bit rate yang didapatkan berdasarkan perhitungan diatas, maka dapat diketahui bahwa nilai bit rate dari sistem kerja VSN dengan menggunakan algoritma efisien mengonsumsi lebih sedikit bit energi dibandingkan dengan sistem kerja VSN yang tidak menggunakan algoritma. Diketahui jumlah bit energinya, maka dapat diketahui nilai efisiensi penghematannya

$$\begin{aligned}
 \text{Penghematan \%} &= \frac{\text{Selisih Daya Bit Energi}}{\text{Jumlah Rata – rata Bit Energi}} \times 100\% \\
 &= \frac{(4,01-3,46)}{4,01} \times 100\% \\
 &= \frac{0,55}{4,01} \times 100\% \\
 &= 13,71 \%
 \end{aligned}$$

Sistem kerja algoritma VSN dapat melakukan penghematan bit energi sebesar 13,71% dibandingkan dengan sistem kerja VSN tanpa menggunakan algoritma.

## **BAB 5**

### **KESIMPULAN DAN SARAN**

Setelah rangkaian penelitian yang telah dilakukan dianalisa maka akan dapat ditarik kesimpulan. Pembahasan dari bab-bab sebelumnya dan kendala-kendala yang terjadi selama pengerjaan tugas akhir ini akan menjadi bahan pertimbangan atau referensi dalam melakukan penelitian pengembangan dari penelitian ini atau penelitian dalam *scoop* topik yang sama.

#### **5.1 Kesimpulan**

Berdasarkan hasil penelitian untuk mengimplementasikan sistem kerja VSN dengan algoritma dapat disimpulkan :

1. Masing-masing node hanya akan melakukan perekaman terhadap kejadian hanya apabila terdapat pergerakan yang melintas.
2. Nilai *threshold* dapat ditentukan sesuai dengan sensitifitas pergerakan yang diinginkan.
3. Dari perbandingan data yang disajikan antara pengimplementasian *Visual Sensor Networks* dengan menggunakan algoritma yang diujicobakan dalam penelitian ini, didapatkan bahwa kualitas resolusi dari keduanya tidak ada perbedaan walaupun menggunakan daya dan energi bit yang berbeda.
4. Kondisi *idle* pada *Wireless Sensor Networks* maupun pada *Visual Sensor Networks* mempunyai konsumsi daya dan energi yang berbeda dengan kondisi selalu melakukan recording.
5. Penggunaan algoritma efisien yang bertujuan untuk melakukan penghematan terhadap konsumsi daya, bit energi, maupun besarnya file yang dihasilkan dari proses perekaman *Visual Sensor Networks*. Hal ini terlihat dari besarnya nilai penghematan yang dilakukan terhadap kondisi tersebut. Penghematan daya bisa dilakukan hingga sekitar  $\pm 19,6\%$  dan penghematan konsumsi bit energi hingga  $\pm 13,71\%$ .

#### **5.2 Saran**

Tugas akhir ini diharapkan dapat diaplikasikan untuk membantu perkembangan teknologi *Visual Sensor Network* yang sedang dalam perkembangan pesat. Di masa yang akan datang, diramalkan semakin banyak aplikasi dan inovasi dari teknologi semacam ini. Khususnya melihat potensi semakin banyaknya kota di Indonesia

menerapkan konsep *smart city* dimana aspek kamera pengawas memegang peranan utama di dalamnya. Selain itu masih ada kekurangan dalam sistem ini, yaitu pada *threshold* yang dipakai dalam sistem ini. *Threshold* yang dipakai harus bisa meminimalisir kemungkinan objek lain selain manusia yang ditangkap agar tidak terjadi perekaman sia-sia dalam prosesnya.

Selain itu dalam pengembangannya, semoga tugas akhir ini bisa dilakukan pengembangan penelitian. Penerapan dengan sistem nirkabel dan menggunakan perangkat keras yang lebih kompleks, seperti *Visual Sensor Networks* dikombinasikan dengan *quadcopter* dan mikroprosesor Raspberry pi.

## DAFTAR PUSTAKA

- [1] M. Al Najjar, M. Ghantous, M. Bayoumi. "Video Surveillance for Sensor Platforms: Algorithms and Architectures", Springer Science+Business Media, United States: Syngress, 2014.
- [2] Krishnamachari B., Murphy A.L., "*Wireless Sensor Networks*, 11th European Conference". Oxford, United Kingdom, 2014.
- [3] Monk S., "*Raspberry Pi Cookbook*," O'Reilly, United States, Dec. 2013.
- [4] Tavli B, Bicakci K, Zilan R, and Barcelo-Ordinas J.M., "A survey of *Visual Sensor Network* platforms," Springer Science+Business Media, United States, 2011.
- [5] Bell C, "*Beginning Sensor Networks with Arduino and Raspberry Pi*". New York, United States, 2013.
- [6] Jawad Ali S., Roy P., "Energy Saving in *Wireless Sensor Networks*," Halmstad University, Sweden, 2008.
- [7] Redondi A., Buranapanichkit D., Cesana M., Tagliasacchi M., and Andreopoulos Y. "Energy Consumption of Visual Sensor Networks: Impact of Spatio-Temporal Coverage". IEEE, United Kingdom, 2014.
- [8] Margi Cintia B., Petkov V., Obraczka K., and Manduchi R. "Characterizing Energy Consumption in a Visual Sensor Network Testbed," IEEE, Santa Cruz, 2006.
- [9] Chow Kit-Y., Lui King-S., and Lam Edmund Y., "Balancing Image Quality and Energy Consumption in Visual Sensor Networks" IEEE, China, 2006.
- [10] Solichin A., Harjoko A., "Metode Background Substraction untuk Deteksi Obyek Pejalan Kaki pada Lingkungan Statis". SNATI, Yogyakarta, 2013.

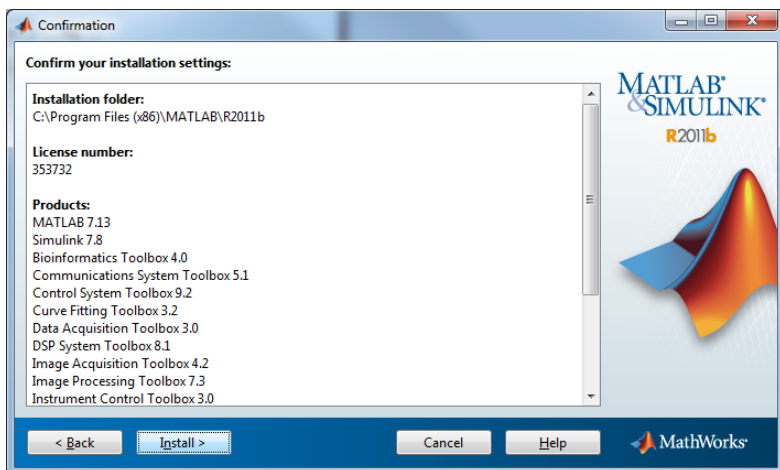
## LAMPIRAN

### A. Instalasi Matlab R2011b

Beberapa situs atau web terkadang menyediakan file master dari instalasi Matlab yang dibagikan/dishare. Banyak yang menyediakan instalasi dari Matlab ini.

Bila kita melakukan instalasi Matlab menggunakan Compact Disk (CD) bisa menggunakan AutoRUN atau pilih “setup”, maka kita akan masuk ke proses instalasi Matlab. Setelah itu bisa melakukan proses instalasi dan dilanjutkan dengan memindahkan file-file yang dibutuhkan untuk menjalankan Matlab.

Saat kita akan melakukan proses instalasi Matlab, diperlukan serial number, dan diakhiri dengan aktivasi setelah Matlab berhasil diinstal. Baca license agreement saat melakukan proses instalasi.



**Gambar Lampiran Proses Instalasi Matlab R2011b**

Matlab akan membuat folder di drive C dengan address C:\Program Files\MATLAB\R2011b\.. Terakhir Matlab akan menampilkan fitur-fitur apa saja yang akan dipasang. Saat proses instalasi selesai, maka kita akan diminta menjalankan langsung program Matlab yang baru saja kita instal atau melakukan proses aktivasi. Jika kita telah memiliki file aktivasi, kita tinggal mengarahkan ke file aktivasi pada komputer atau laptop kita. Jika icon atau shortcut Matlab tidak

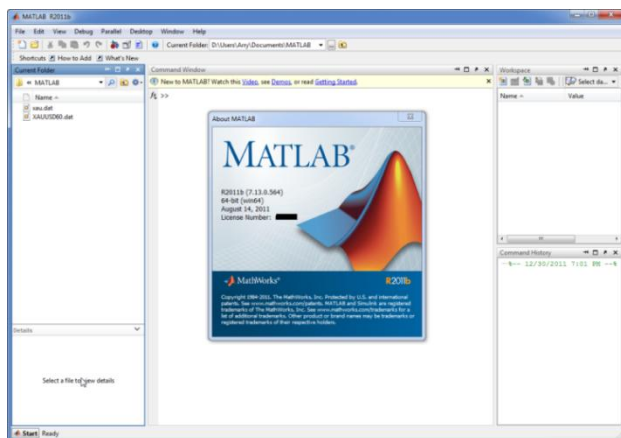
muncul pada desktop kita, bukan berarti kita gagal melakukan instalasi Matlab, bisa dengan cara explore di lokasi C:\Program Files\MATLAB\R2011b\bin. Kita akan menjumpai Matlab.exe, lalu bisa dibuat shortcut ke desktop.

Banyak perangkat lunak atau software yang mirip dengan Matlab tetapi open source seperti Scilab, Octave, Euler Math Toolbox, dan FreeMat. Namun, tentu saja sedikit kurang fleksibel dibandingkan dengan Matlab.

Matlab lebih fleksibel karena Matlab merupakan bahasa pemrograman tingkat tinggi berbasis pada matriks, sering kita gunakan untuk teknik komputasi numerik, yang kita gunakan untuk menyelesaikan masalah-masalah yang melibatkan operasi matematika elemen, matriks, optimasi, aproksimasi dan lain sebagainya.

Maka dari fungsi Matlab yang seperti dijelaskan diatas itulah banyak pengguna yang memanfaatkan Matlab sebagai alat untuk melakukan operasi pada bidang

- Matematika dan komputasi
- Pengembangan dan algoritma
- Pemrograman modeling, simulasi, dan pembuatan prototipe
- Analisis data, eksplorasi dan visualisasi
- Analisis numerik dan statistik
- Pengembangan aplikasi teknik



**Gambar Lampiran Start-up pada Matlab R2011b**

## B. Instalasi OpenCV

Untuk melakukan pembuatan dan perencanaan algoritma pada kamera dapat melakukan bantuan aplikasi OpenCV. OpenCV termasuk salah satu aplikasi yang populer dalam dunia pemrograman menggunakan kamera dan pendeteksian visual lainnya. Banyak referensi yang dapat membantu dalam penginstallan OpenCV.

Setelah OpenCV dapat terinstall dengan sempurna, maka kita dapat memulai proses coding dengan membuka Microsoft Visual Studio. Beberapa pengimplementasian yang dapat dilakukan dari Computer Vision antara lain Face Recognition, Face Detection, Face/Object Tracking, Road Tracking, dan banyak fungsi lainnya.

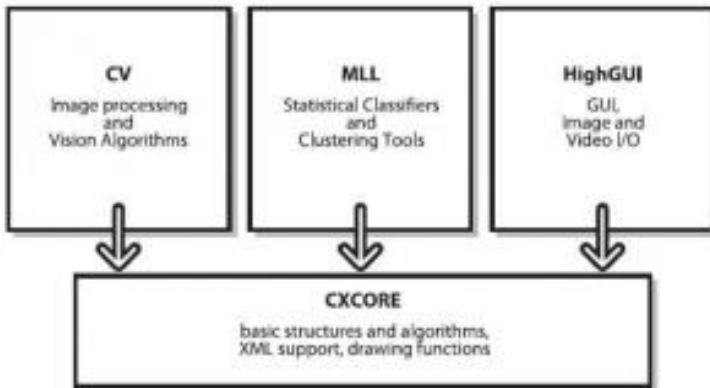
OpenCV sendiri terdiri dari 5 library, yaitu:

CV : untuk algoritma Image Processing dan Vision

ML : untuk machine learning library

Highgui : untuk GUI, Image dan Video I/O

CXCore : untuk struktur data, support XML dan fungsi-fungsi grafis  
Dan CvAux



**Gambar Lampiran Struktur dan Konten OpenCV**

## C. Instalasi C++

Serupa dengan bahasa pemrograman lainnya, bahasa pemrograman C++ mudah untuk dilakukan instalasinya. Bahasa pemrograman ini masih banyak digunakan pada beberapa disiplin ilmu yang menerapkan ilmu tentang komputer.



Untuk mengawali penginstalan program C++ ini, pertama dibutuhkan perangkat lunak pembantu dalam proses instalasi DosBox. Setelah Dosbox berhasil diinstal, selanjutnya kita buat folder misalkan kita beri nama “Viktori” (C:\Viktori\). Download dan ekstrak ke folder TC Viktori (C:\Viktori\). Lalu jalankan DosBox. Selanjutnya masukan perintah

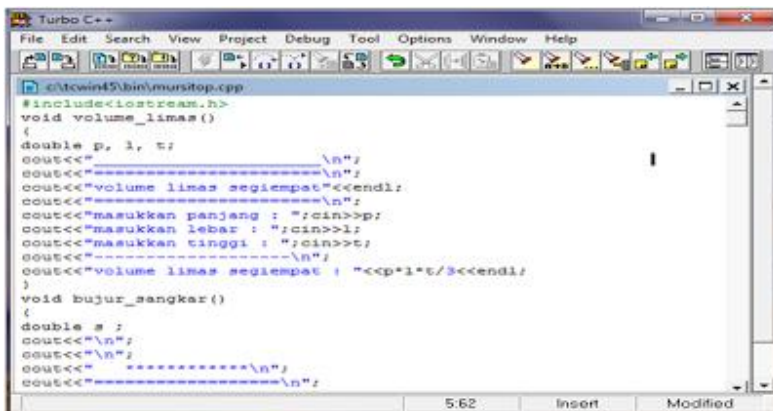
```
[Z]: mount dc:\Viktori\  
(Folder TC hadir di dalam folder Viktori)
```

Sekarang kita harus mendapatkan pesan yang mengatakan Drive D-mount sebagai direktori lokal C:\Viktori\ . Selanjutnya kita bisa memasukkan perintah

```
Cd tc  
Cd bin  
tc atau tc.exe [Ini memulai Viktori C++]
```

Jika program sudah berhasil running, kita ubah direktori, caranya klik options, lalu klik directories yaitu

D:\TC\include dan D:\TC\librespectively



**Gambar Lampiran Tampilan Pemrograman C++**

#### D. Program Simulasi Sederhana dengan Menggunakan Matlab R2011b

```
clc;
close all;
clear;

%Read Background Image
Background=imread('background.jpg');
%Read Current Frame
CurrentFrame=imread('original.jpg');
%Display Background and Foreground
% subplot(2,2,1);imshow(Background);title('BackGround');
% subplot(2,2,2);imshow(CurrentFrame);title('Current Frame');

Background_gray=rgb2gray(Background);
CurrentFrame_gray=rgb2gray(CurrentFrame);

[rows columns]=size(Background_gray);

%Convert to Binary Image
for i=1:rows
    for j=1:columns
        if (Background(i,j)-CurrentFrame(i,j)) > 75
            BinaryImage(i,j)=1;
        else
            BinaryImage(i,j)=0;
        end
    end
end

SE = strel('arbitrary',eye(5));
BinaryImage_erode = imerode(BinaryImage,SE);
BinaryImage_dilate = imdilate(BinaryImage_erode,SE);

BinaryImage_dilate_edit = bwareaopen(BinaryImage_dilate, 2000);

[B,L,N,A] = bwboundaries(BinaryImage_dilate);
```

```

%[B,L,N,A] = bwboundaries(BinaryImage_dilate_edit);

for k=1:length(B)
    [m,n] = size(B{k});
    fprintf('Keliling blob %d adalah %d pixel \n', k, m);
    if m > 200
        fprintf('ADA PERGERAKAN \n');
    end
end

figure, imshow(Background);title('BackGround');
figure, imshow(CurrentFrame);title('Current Frame');
figure, imshow(BinaryImage);title('BackGround Subtraction');
figure, imshow(BinaryImage_erode);title('BackGround Erode');
figure, imshow(BinaryImage_dilate);title('BackGround Dilate');
figure, imshow(BinaryImage_dilate_edit);title('BackGround Dilate
Edit');
[m,n]=size(BinaryImage_dilate);
a=[];
b=[];
for i=1:m;
    for j=1:n;
        if BinaryImage_dilate(i,j)==1;
            a=[a i];
            b=[b j];
        end
    end
end
loc=size(a,2);
pointy=a(loc);
pointx=b(loc);

```

## E. Program Node Server Tanpa Menggunakan Algoritma

```

#include<opencv2/opencv.hpp>
#include<iostream>
#include<vector>
#include <pthread.h>
#include <stdio.h>

```

```

#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

using namespace cv;
using namespace std;

pthread_mutex_t mutex1 = PTHREAD_MUTEX_INITIALIZER;
int perintah = 0;

void * rekam(void * ptr )
{
    Mat frame, gray_frame;
    Mat cap_img, back, diff;
    Mat kernel_dil = getStructuringElement(MORPH_RECT, Size(3,3));
    Mat kernel_ero = getStructuringElement(MORPH_RECT,
    Size(5,5));

    //ambil frame dari video
    VideoCapture cap(0);

    cap.set(CV_CAP_PROP_FRAME_WIDTH, 320);
    cap.set(CV_CAP_PROP_FRAME_WIDTH, 240);

    if(!cap.isOpened()){
        cout << "Error opening video stream or file" << endl;
        //return -1;
    }

    //inisialisasi video
    int frame_width= cap.get(CV_CAP_PROP_FRAME_WIDTH);
    int frame_height=
    cap.get(CV_CAP_PROP_FRAME_HEIGHT);

```

```

VideoWriter
video("VideoNonAlgoritma.avi",CV_FOURCC('M','J','P','G'),15,
Size(frame_width/2,frame_height/2),true);

while(1)
{
    if (perintah == 0)
        continue;
    else if (perintah == 2)
        break;
        //cap >> back;
    //cvtColor(back, back, CV_RGB2GRAY);

    cap >> frame;
    //cvtColor(frame, gray_frame, CV_RGB2GRAY);

resize(frame,frame,Size(frame_width/2,frame_height/2),0,0,INTER_CU
BIC);
    video.write(frame);


    imshow("frame", frame);
    char c = cvWaitKey(1);

    if (c == 27) break;

}
//return 0;
}

int main(int argc, char *argv[])
{
    int sockfd, newsockfd, portno;
    socklen_t clilen;
    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int n;

```

```

pthread_t threadid;
pthread_create(&threadid, NULL, rekam, NULL);


sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0)
    printf("ERROR opening socket\n");
bzero((char *) &serv_addr, sizeof(serv_addr));
portno = 4545;
serv_addr.sin_family = AF_INET;
serv_addr.sin_addr.s_addr = INADDR_ANY;
serv_addr.sin_port = htons(portno);
if (bind(sockfd, (struct sockaddr *) &serv_addr,
    sizeof(serv_addr)) < 0)
    printf("ERROR on binding\n");
listen(sockfd,5);
clilen = sizeof(cli_addr);
newsockfd = accept(sockfd,
    (struct sockaddr *) &cli_addr,
    &clilen);
if (newsockfd < 0)
    printf("ERROR on accept\n");

while(1)
{
    int _perintah;
    bzero(buffer,256);
    n = read(newsockfd,buffer,255);
    _perintah = atoi(buffer);

    pthread_mutex_lock(&mutex1);
    perintah = _perintah;
    pthread_mutex_unlock(&mutex1);

    if (_perintah == 2)
        break;
}
close(newsockfd);
close(sockfd);

```

```
}
```

## **F. Program Node Client Tanpa Menggunakan Algoritma**

```
#include<opencv2/opencv.hpp>
#include<iostream>
#include<vector>
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

using namespace cv;
using namespace std;

pthread_mutex_t mutex1 = PTHREAD_MUTEX_INITIALIZER;
int perintah = 0;

void * rekam(void * ptr )
{
    Mat frame, gray_frame;
    Mat cap_img, back, diff;
    Mat kernel_dil = getStructuringElement(MORPH_RECT, Size(3,3));
    Mat kernel_ero = getStructuringElement(MORPH_RECT,
    Size(5,5));

    //ambil frame dari video
    VideoCapture cap(0);

    if(!cap.isOpened()){
        cout << "Error opening video stream or file" << endl;
        //return -1;
    }
}
```

```

//inisialisasi video
int frame_width= cap.get(CV_CAP_PROP_FRAME_WIDTH);
int frame_height=
cap.get(CV_CAP_PROP_FRAME_HEIGHT);
VideoWriter
video("VideoNonAlgoritma.avi",CV_FOURCC('M','J','P','G'),15,
Size(frame_width/2,frame_height/2),true);

while(1)
{
    if (perintah == 0)
        continue;
    else if (perintah == 2)
        break;
        //cap >> back;
    //cvtColor(back, back, CV_RGB2GRAY);

    cap >> frame;
    //cvtColor(frame, gray_frame, CV_RGB2GRAY);

    resize(frame,frame,Size(frame_width/2,frame_height/2),0,0,INTER_CU
    BIC);
    video.write(frame);

        imshow("frame", frame);
        char c = cvWaitKey(1);
        if (c == 27) break;

}
//return 0;
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
    struct sockaddr_in serv_addr;
    struct hostent *server;

```



```

char buffer[256];

pthread_t threadid;
pthread_create(&threadid, NULL, rekam, NULL);

portno = 4545;
sockfd = socket(AF_INET, SOCK_STREAM, 0);
if (sockfd < 0)
    printf("ERROR opening socket\n");

server = gethostbyname("192.168.1.69");
if (server == NULL)
{
    fprintf(stderr, "ERROR, no such host\n");
    exit(0);
}

bzero((char *) &serv_addr, sizeof(serv_addr));
serv_addr.sin_family = AF_INET;
bcopy((char *)server->h_addr,
(char *)&serv_addr.sin_addr.s_addr,
server->h_length);
serv_addr.sin_port = htons(portno);

if (connect(sockfd,(struct sockaddr *)
&serv_addr,sizeof(serv_addr)) < 0)
    printf("ERROR connecting\n");

while(1)
{
    int _perintah;
    cin >> _perintah;

    bzero(buffer,256);
    sprintf(buffer, "%d", _perintah);
    n = write(sockfd,buffer,strlen(buffer));

```

```

        pthread_mutex_lock(&mutex1);
        perintah = _perintah;
        pthread_mutex_unlock(&mutex1);
    if (perintah == 2)
        break;
    }
}

```

### **G. Program Node Server dengan Menggunakan Algoritma**

```

#include<opencv2/opencv.hpp>
#include<iostream>
#include<vector>
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

using namespace cv;
using namespace std;

pthread_mutex_t mutex1 = PTHREAD_MUTEX_INITIALIZER;
int perintah = 0;

void * rekam(void * ptr )
{
    Mat frame, gray_frame;
    Mat cap_img, back, diff;
    Mat kernel_dil = getStructuringElement(MORPH_RECT, Size(3,3));
    Mat kernel_ero = getStructuringElement(MORPH_RECT,
    Size(5,5));

    //ambil frame dari video
    VideoCapture cap(0);

```

```

        int frame_width=
cap.get(CV_CAP_PROP_FRAME_WIDTH);
        int frame_height=
cap.get(CV_CAP_PROP_FRAME_HEIGHT);
        int trsh =          500;

//VideoCapture cap(0);
if(!cap.isOpened()){
    cout << "Error opening video stream or file" << endl;
}

//inisialisasi video

VideoWriter
video1("VideoAlgoritma.avi",CV_FOURCC('M','J','P','G'),15,
Size(frame_width/2,frame_height/2),true);

while(1)
{
    if (perintah == 0)
        continue;
    else if (perintah == 2)
        break;

        cap >> back;
        cvtColor(back, back, CV_RGB2GRAY);

        cap >> frame;
        cvtColor(frame, gray_frame, CV_RGB2GRAY);

        ///background subtraction
        absdiff(back, gray_frame, diff);
        threshold(diff, diff, 60, 255,
CV_THRESH_BINARY+THRESH_OTSU);
        erode(diff, diff, kernel_ero);
        //dilate(diff, diff, kernel_dil);

```

```

    int nilai = countNonZero(diff);

    if (nilai > trsh)
    {
        //kompresi resolusi

        resize(frame,frame,Size(frame_width/2,frame_height/2),0,0,INTER_CUBIC);

        video1.write(frame);
    }
    imshow("frame", frame);

    char c = cvWaitKey(1);
    if (c == 27) break;

}
//return 0;
}

int main(int argc, char *argv[])
{
    int sockfd, newsockfd, portno;
    socklen_t clilen;
    char buffer[256];
    struct sockaddr_in serv_addr, cli_addr;
    int n;

    pthread_t threadid;
    pthread_create(&threadid, NULL, rekam, NULL);

    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        printf("ERROR opening socket\n");
    bzero((char *) &serv_addr, sizeof(serv_addr));
    portno = 4545;
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(portno);

```

```

if (bind(sockfd, (struct sockaddr *) &serv_addr,
    sizeof(serv_addr)) < 0)
    printf("ERROR on binding\n");
listen(sockfd,5);
clilen = sizeof(cli_addr);
newsockfd = accept(sockfd,
    (struct sockaddr *) &cli_addr,
    &clilen);
if (newsockfd < 0)
    printf("ERROR on accept\n");

while(1)
{
    int _perintah;
    bzero(buffer,256);
    n = read(newsockfd,buffer,255);
    _perintah = atoi(buffer);

    pthread_mutex_lock(&mutex1);
    perintah = _perintah;
    pthread_mutex_unlock(&mutex1);

    if (_perintah == 2)
        break;
}
close(newsockfd);
close(sockfd);
}

```

## H. Program Node Client dengan Menggunakan Algoritma

```

#include<opencv2/opencv.hpp>
#include<iostream>
#include<vector>
#include <pthread.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <string.h>
#include <sys/types.h>

```

```

#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>

using namespace cv;
using namespace std;

pthread_mutex_t mutex1 = PTHREAD_MUTEX_INITIALIZER;
int perintah = 0;

void * rekam(void * ptr )
{
    Mat frame, gray_frame;
    Mat cap_img, back, diff;
    Mat kernel_dil = getStructuringElement(MORPH_RECT, Size(3,3));
    Mat kernel_ero = getStructuringElement(MORPH_RECT,
Size(5,5));

    //ambil frame dari video
    VideoCapture cap(0);

    cap.set(CV_CAP_PROP_FRAME_WIDTH, 320);
    cap.set(CV_CAP_PROP_FRAME_WIDTH, 240);

    int frame_width=
cap.get(CV_CAP_PROP_FRAME_WIDTH);
    int frame_height=
cap.get(CV_CAP_PROP_FRAME_HEIGHT);
    int trsh =      500;

    //VideoCapture cap(0);
    if(!cap.isOpened()){
        cout << "Error opening video stream or file" << endl;
    }

    //inisialisasi video

```

```

VideoWriter
video1("VideoAlgoritma.avi",CV_FOURCC('M','J','P','G'),15,
Size(frame_width/2,frame_height/2),true);

while(1)
{
    if (perintah == 0)
        continue;
    else if (perintah == 2)
        break;

        cap >> back;
    cvtColor(back, back, CV_RGB2GRAY);

    cap >> frame;
    cvtColor(frame, gray_frame, CV_RGB2GRAY);

    //background subtraction
    absdiff(back, gray_frame, diff);
        threshold(diff, diff, 60, 255,
CV_THRESH_BINARY+THRESH_OTSU);
    erode(diff, diff, kernel_ero);
        //dilate(diff, diff, kernel_dil);

    int nilai = countNonZero(diff);

    if (nilai > trsh)
    {
        //kompresi resolusi

        resize(frame,frame,Size(frame_width/2,frame_height/2),0,0,IN
TER_CUBIC);
        video1.write(frame);
    }
    imshow("frame", frame);

    char c = cvWaitKey(1);
    if (c == 27) break;

```

```

    }
    //return 0;
}

int main(int argc, char *argv[])
{
    int sockfd, portno, n;
        struct sockaddr_in serv_addr;
        struct hostent *server;
        char buffer[256];

        pthread_t threadid;
        pthread_create(&threadid, NULL, rekam, NULL);

    portno = 4545;
    sockfd = socket(AF_INET, SOCK_STREAM, 0);
    if (sockfd < 0)
        printf("ERROR opening socket\n");

        server = gethostbyname("192.168.1.69");
        if (server == NULL)
        {
            fprintf(stderr, "ERROR, no such host\n");
            exit(0);
        }

        bzero((char *) &serv_addr, sizeof(serv_addr));
        serv_addr.sin_family = AF_INET;
        bcopy((char *)server->h_addr,
            (char *)&serv_addr.sin_addr.s_addr,
            server->h_length);
        serv_addr.sin_port = htons(portno);

        if (connect(sockfd,(struct sockaddr *)
&serv_addr,sizeof(serv_addr)) < 0)
            printf("ERROR connecting\n");

    while(1)

```



```
{  
    int _perintah;  
    cin >> _perintah;  
  
    bzero(buffer,256);  
    sprintf(buffer, "%d", _perintah);  
    n = write(sockfd,buffer,strlen(buffer));  
  
    pthread_mutex_lock(&mutex1);  
    perintah = _perintah;  
    pthread_mutex_unlock(&mutex1);  
    if (perintah == 2)  
        break;  
}}
```

**TE141599 TUGAS AKHIR – 4 SKS**

Nama Mahasiswa : Hananto Agung Baskoro  
Nomor Pokok : 2211100150  
Bidang Studi : Teknik Telekomunikasi Multimedia  
Tugas Diberikan : Semester Genap Th. 2014/2015  
Dosen Pembimbing : 1. Eko Setijadi, ST., MT., Ph.D  
2. Dr. Ir. Wirawan, DEA.

16 FEB 2015

Judul Tugas Akhir : **Visual Sensor Network dengan Algoritma yang Hemat Energi**  
(*Visual Sensor Network with Energy Saving Algorithm*)

**Uraian Tugas Akhir :**

Jaringan sensor nirkabel memiliki potensi yang sangat bagus untuk diterapkan pada banyak aspek seperti pada militer, penanggulangan bencana, penelitian tempat berbahaya, peningkatan keamanan dalam suatu sistem kerja, dan masih banyak lagi. Salah satu jenis dari jaringan sensor nirkabel adalah Visual Sensor Network (VSN). Dimana Visual Sensor Network ini adalah pendeteksi citra gambar maupun citra video. Namun dalam prinsip kerjanya semakin besar bit yang dihasilkan pada kamera VSN, maka energi yang dikeluarkan juga akan berbanding lurus yaitu semakin besar. Maka dari itu perlu suatu algoritma yang hemat energi yang nantinya diterapkan pada VSN yang menjangkau suatu daerah cakupan tertentu.

Pada tugas akhir ini akan dilakukan pemberian pengkodean algoritma pada Visual Sensor Network agar mendapatkan hasil VSN yang dapat mendapatkan citra gambar resolusi menengah sampai tinggi dengan konsumsi energi yang rendah. Selain itu dari tugas akhir ini akan menghasilkan data simulasi dari MATLAB dan juga data dari percobaan implementasi platform dari alat laboratorium.

Kata Kunci : *Visual Sensor Network, Energy Consumption, Node*

Dosen Pembimbing I,



**Eko Setijadi, ST., MT., Ph.D.**  
NIP. 197210012003121002

Dosen Pembimbing II,



**Dr. Ir. Wirawan, DEA.**  
NIP. 196311091989031011

Mengetahui,  
Jurusan Teknik Elektro FTI – ITS  
Ketua,



**Dr. Tri Arief Sardiono, ST., MT.**  
NIP. 197002121995121001

Menyetujui,  
Bidang Studi Telekomunikasi Multimedia  
Koordinator,



**Dr. Ir. Endroyono, DEA**  
NIP. 1965040419910210

## BIOGRAFI PENULIS



Hananto Agung Baskoro, lahir di Jakarta – DKI Jakarta pada tanggal 7 Juni 1993. Penulis merupakan anak kedua dari tiga bersaudara. Pada tahun 2005, penulis menamatkan pendidikan Sekolah Dasar di SD Bani Saleh 1 Bekasi, kemudian melanjutkan pendidikan di SMP Al-Azhar 8 Bekasi dan selesai tepat pada tahun 2008. Penulis melanjutkan pendidikan Sekolah Menengah Umum di SMA Negeri 81 Jakarta hingga lulus pada tahun 2011. Dengan anugerah Allah SWT, penulis dapat melanjutkan studi di salah satu universitas terkemuka di Indonesia.

Institut Teknologi Sepuluh Nopember dengan mengambil Jurusan Teknik Elektro. Penulis mengambil konsentrasi Bidang Studi Telekomunikasi Multimedia dan aktif sebagai asisten laboratorium di 2 laboratorium yaitu Laboratorium Jaringan dan Laboratorium Multimedia. Sewaktu kuliah penulis juga aktif dalam kegiatan kemahasiswaan Himpunan Mahasiswa Teknik Elektro ITS serta berpartisipasi dalam seluruh jenjang kegiatan pelatihan LKMM ITS dari tingkat pra dasar sampai tingkat lanjut. Email: [hananto.baskoro@gmail.com](mailto:hananto.baskoro@gmail.com)